Entitybroker-api module

Package: org.sakaiproject.entitybroker

Interface: DeveloperHelperClass

Contains methods which are likely to be helpful to developers who are implementing entity providers in Sakai and working with references.

Methods:

### cloneBean

<T> T cloneBean(T bean,

int maxDepth,

java.lang.String[] propertiesToSkip)

Deep clone a bean (object) and all the values in it into a brand new object of the same type, this will traverse the bean and will make new objects for all non-null values contained in the object, the level indicates the number of contained objects to traverse and clone, setting this to zero will only clone basic type values in the bean, setting this to one will clone basic fields, references, and collections in the bean, etc.  
This is mostly useful for making a copy of a hibernate object so it will no longer be the persistent object with the hibernate proxies and lazy loading

Type Parameters:

T -

Parameters:

bean - any java bean, this can also be a list, map, array, or any simple object, it does not have to be a custom object or even a java bean, also works with apache beanutils DynaBeans

maxDepth - the number of objects to follow when traveling through the object and copying the values from it, 0 means to only copy the simple values in the object, any objects will be ignored and will end up as nulls, 1 means to follow the first objects found and copy all of their simple values as well, and so forth

propertiesToSkip - the names of properties to skip while cloning this object, this only has an effect on the bottom level of the object, any properties found on child objects will always be copied (if the maxDepth allows)

Returns:

the clone of the bean

Throws:

java.lang.IllegalArgumentException - if there is a failure cloning the bean

### convert

<T> T convert(java.lang.Object object,

java.lang.Class<T> type)

Converts an object to any other object if possible

Type Parameters:

T -

Parameters:

object - any object

type - any class type that you want to try to convert the object to

Returns:

the converted value (allows null to pass through except in the case of primitives which become the primitive default)

Throws:

java.lang.UnsupportedOperationException - if the conversion cannot be completed

### copyBean

void copyBean(java.lang.Object orig,

java.lang.Object dest,

int maxDepth,

java.lang.String[] fieldNamesToSkip,

boolean ignoreNulls)

Deep copies one bean (object) into another, this is primarily for copying between identical types of objects but it can also handle copying between objects which are quite different, this does not just do a reference copy of the values but actually creates new objects in the current classloader and traverses through all properties of the object to make a complete deep copy

Parameters:

original - the original object to copy from

destination - the object to copy the values to (must have the same fields with the same types)

maxDepth - the number of objects to follow when traveling through the object and copying the values from it, 0 means to only copy the simple values in the object, any objects will be ignored and will end up as nulls, 1 means to follow the first objects found and copy all of their simple values as well, and so forth

fieldNamesToSkip - the names of fields to skip while cloning this object, this only has an effect on the bottom level of the object, any fields found on child objects will always be copied (if the maxDepth allows)

ignoreNulls - if true then nulls are not copied and the destination retains the value it has, if false then nulls are copied and the destination value will become a null if the original value is a null

Throws:

java.lang.IllegalArgumentException - if the copy cannot be completed because the objects to copy do not have matching fields or types

### decodeData

java.util.Map<java.lang.String,java.lang.Object> decodeData(java.lang.String data,

java.lang.String format)

Decode a string of a specified format into a java map of simple objects   
Returned map can be fed into [populate(Object, Map)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\DeveloperHelperService.html#populate%28java.lang.Object,%20java.util.Map%29) if you want to convert it into a known object type   
Types are likely to require conversion as guesses are made about the right formats, use of the [convert(Object, Class)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\DeveloperHelperService.html#convert%28java.lang.Object,%20java.lang.Class%29) method is recommended   
Current formats supported: JSON, XML

Parameters:

data - encoded data

format - the format of the encoded data (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html))

Returns:

a map containing all the data derived from the encoded data

Throws:

java.lang.UnsupportedOperationException - if the data cannot be decoded

java.lang.IllegalArgumentException - if the data cannot be decoded because there is no decoder for that format

### encodeData

java.lang.String encodeData(java.lang.Object data,

java.lang.String format,

java.lang.String name,

java.util.Map<java.lang.String,java.lang.Object> properties)

Encode data into a given format, can handle any java object, note that unsupported formats and invalid data will result in an exception   
Current formats supported: JSON, XML, HTML

Parameters:

data - the data to encode

format - the format to use for output (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html))

name - (optional) the name to use for the encoded data (e.g. root node for XML)

properties - (optional) extra properties to add into the encoding, ignored if encoded object is not a map or bean

Returns:

the encoded string in the requested format

Throws:

java.lang.UnsupportedOperationException - if the data cannot be encoded

java.lang.IllegalArgumentException - if the format requested cannot be encoded because there is no encoder

### entityExists

boolean entityExists(java.lang.String reference)

Convenience method from [EntityBroker](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityBroker.html)  
Check if an entity exists by the globally unique reference string, (the global reference string will consist of the entity prefix and any local ID). If no [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) for the reference is found which implements [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html), this method will return true by default, in other words, this cannot determine if a legacy entity exists, only a new entity

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments (normally the id at least)

Returns:

true if the entity exists, false otherwise

### fetchEntity

java.lang.Object fetchEntity(java.lang.String reference)

Convenience method from [EntityBroker](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityBroker.html)  
Fetches a concrete object representing this entity reference; either one from the [Resolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Resolvable.html) capability if implemented by the responsible [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), or else from the underlying legacy Sakai entity system

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments

Returns:

an object which represents the entity or null if none can be found

### fireEvent

void fireEvent(java.lang.String eventName,

java.lang.String reference)

Convenience method from [EntityBroker](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityBroker.html)  
Fire an event to Sakai with the specified name, targetted at the supplied reference, which should be a reference to an existing entity managed by this broker  
NOTE: This will allow events to be fired for references without a broker or invalid references

Parameters:

eventName - a string which represents the name of the event (e.g. announcement.create), cannot be null or empty

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments, cannot be null or empty

### getConfigurationSetting

<T> T getConfigurationSetting(java.lang.String settingName,

T defaultValue)

Retrieves settings from the configuration service (sakai.properties)

Parameters:

settingName - the name of the setting to retrieve, Should be a string name: e.g. auto.ddl, mystuff.config, etc. OR one of the SETTING constants (e.g [SETTING\_AUTO\_DDL](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\DeveloperHelperService.html#SETTING_AUTO_DDL))

defaultValue - a specified default value to return if this setting cannot be found, NOTE: You can set the default value to null but you must specify the class type in parens

Returns:

the value of the configuration setting OR the default value if none can be found

### getCurrentLocale

java.util.Locale getCurrentLocale()

Returns:

the Locale for the current user or the system set locale

### getCurrentLocationId

java.lang.String getCurrentLocationId()

Returns:

the entity id of the current location for the current session (represents the current site/group of the current user in the system)   
recommend you use [getCurrentLocationReference()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\DeveloperHelperService.html#getCurrentLocationReference%28%29) instead

### getCurrentLocationReference

java.lang.String getCurrentLocationReference()

Returns:

the entity reference (/site/{siteId}) of the current location for the current session (represents the current site/group of the current user in the system)

### getCurrentToolReference

java.lang.String getCurrentToolReference()

Returns:

the entity reference of the current active tool for the current session (represents the tool that is currently being used by the current user in the system)

### getCurrentUserId

java.lang.String getCurrentUserId()

Get the user entity id (e.g. not eid or username) of the current user if there is one, this is not equivalent to the current user reference   
recommend you use [getCurrentUserReference()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\DeveloperHelperService.html#getCurrentUserReference%28%29) instead

Returns:

the user id OR null if there is no current user (or the user is anonymous)

### getCurrentUserReference

java.lang.String getCurrentUserReference()

Get the user entity reference (e.g. /user/{userId} - not id, eid, or username) of the current user if there is one, this is not equivalent to the current user id

Returns:

the user entity reference (e.g. /user/{userId} - not id, eid, or username) OR null if none

### getEntityReferencesForUserAndPermission

java.util.Set<java.lang.String> getEntityReferencesForUserAndPermission(java.lang.String userReference,

java.lang.String permission)

Find the entity references which a user has a specific permission in, this is most commonly used to get the list of sites which a user has a permission in but it will work for any entity type which uses Sakai permissions

Parameters:

userReference - the user entity reference (e.g. /user/{userId} - not id, eid, or username)

permission - a permission string constant

Returns:

a set of entity references - a globally unique reference to an entity, consists of the entity prefix and optional segments (normally the id at least)

### getEntityURL

java.lang.String getEntityURL(java.lang.String reference,

java.lang.String viewKey,

java.lang.String extension)

Convenience method from [EntityBroker](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityBroker.html)  
Get the full absolute URL to the entity view defined by these params, this will fail-safe to a direct URL to an entity space URL if that is all that is available, this will use the default entity URL template associated with the viewKey and include an optional extension if specified (these will be inferred if they are missing)

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optionally the local id

viewKey - the specific view type to get the URL for, use the VIEW\_\* constants from [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) (e.g. [EntityView.VIEW\_LIST](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#VIEW_LIST)), can be null to determine the key automatically

extension - the optional extension to add to the end which defines the expected data which is returned, use constants in [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html) (e.g. [Formats.XML](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html#XML)), can be null to use no extension, default is assumed to be html if none is set

Returns:

the full URL string to a specific entity or space, (e.g. http://server/direct/prefix/id)

### getLocationIdFromRef

java.lang.String getLocationIdFromRef(java.lang.String locationReference)

Parameters:

locationReference - the location reference (/site/{siteId})

Returns:

the location id

### getLocationReferenceURL

java.lang.String getLocationReferenceURL(java.lang.String locationReference)

Parameters:

locationReference - an entity reference to a location (e.g. /site/siteId)

Returns:

the full URL to a location (e.g. http://server:port/portal/site/siteId)

Throws:

java.lang.IllegalArgumentException - if this reference does not appear to be valid

### getMessage

java.lang.String getMessage(java.lang.String prefix,

java.lang.String messageKey)

Gets messages from the entity message bundles (the entities which implement [Describeable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Describeable.html) or a sub interface),

Parameters:

prefix - the string which represents a type of entity handled by an entity provider

messageKey - the message bundle property key

Returns:

the property value string OR null if none found

### getPortalURL

java.lang.String getPortalURL()

Returns:

the full portal URL as Sakai understands it (e.g. http://server:port/portal)

### getServerURL

java.lang.String getServerURL()

Returns:

the full server base URL (e.g. http://server:port)

### getStartingLocationReference

java.lang.String getStartingLocationReference()

Returns:

the entity reference of the location which is the main starting point for the system (in Sakai this is probably the reference to the gateway site)

### getToolData

[SakaiToolData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\SakaiToolData.html) getToolData(java.lang.String toolRegistrationId,

java.lang.String locationReference)

Parameters:

toolRegistrationId - this is the id string from the Sakai tool registration XML file (i.e. sakai.mytool.xml) and will probably look something like "sakai.mytool"

locationReference - (optional) an entity reference to a location (e.g. /site/siteId) OR null if it should be for the current site

Returns:

an object which contains data about a tool

Throws:

java.lang.IllegalArgumentException - if any parameters are invalid or a tool with this toolRegistrationId cannot be located in the given locatio

### getToolIdFromToolRef

java.lang.String getToolIdFromToolRef(java.lang.String toolReference)

Translate a tool entity reference into a tool Id

Parameters:

toolReference - the entity reference of a tool (e.g. /tool/{toolId})

Returns:

the toolId (needed for other Sakai API operations)

### getToolViewURL

java.lang.String getToolViewURL(java.lang.String toolRegistrationId,

java.lang.String localView,

java.util.Map<java.lang.String,java.lang.String> parameters,

java.lang.String locationReference)

Generate a URL to a tool which will work from anywhere and can carry parameters with it  
NOTE: you should set the A tag target="\_top" if you are inside an existing tool iFrame

Parameters:

toolRegistrationId - this is the id string from the Sakai tool registration XML file (i.e. sakai.mytool.xml) and will probably look something like "sakai.mytool"

localView - (optional) the local URL of the view/page to navigate to within the tool OR null to go to the starting view/page, examples: /view, /page.jsp, /path/to/someview, make sure you include the leading slash ("/")

parameters - (optional) a map of parameters to include in the URL and send along to the tool (these will be turned into GET parameters in the URL), the map should contain parameterName -> parameterValue (e.g. "thing" -> "value")

locationReference - (optional) an entity reference to a location (e.g. /site/siteId) OR null if it should be for the current site

Returns:

a full URL to a tool (e.g. http://server:port/portal/site/siteId/page/pageId?toolstate-toolpid=/view?thing=value)

Throws:

java.lang.IllegalArgumentException - if any parameters are invalid or a tool with this toolRegistrationId cannot be located in the given location

### getUserHomeLocationReference

java.lang.String getUserHomeLocationReference(java.lang.String userReference)

Get the entity reference of the location of a user's workspace/homespace

Parameters:

userReference - the user entity reference (e.g. /user/{userId} - not id, eid, or username)

Returns:

the entity reference of the location OR null if it cannot be generated

### getUserHomeLocationURL

java.lang.String getUserHomeLocationURL(java.lang.String userReference)

Parameters:

userReference - the user entity reference (e.g. /user/{userId} - not id, eid, or username)

Returns:

the full URL to a user's workspace/homespace (e.g. http://server:port/portal/~someuser)

Throws:

java.lang.IllegalArgumentException - if this user reference does not appear to be valid

### getUserIdFromRef

java.lang.String getUserIdFromRef(java.lang.String userReference)

Translate the user entity reference into a userId

Parameters:

userReference - the user entity reference (e.g. /user/{userId} - not id, eid, or username)

Returns:

the userId as extracted from this user entity reference (needed for some Sakai API operations)

### getUserReferencesForEntityReference

java.util.Set<java.lang.String> getUserReferencesForEntityReference(java.lang.String reference,

java.lang.String permission)

Get the user references which have the given permission in the given entity reference, this is most commonly used to get the users which have a permission in a site but it should work for any entity type which uses Sakai permissions

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments (normally the id at least)

permission - a permission string constant

Returns:

a set of user entity references (e.g. /user/{userId} - not id, eid, or username)

### getUserRefFromUserEid

java.lang.String getUserRefFromUserEid(java.lang.String userEid)

Translate the user EID (username/loginname typicaly) into a user reference

Parameters:

userEid - the external user Id (probably the loginname or username)

Returns:

the user entity reference (e.g. /user/{userId}) OR null if the eid is invalid

### getUserRefFromUserId

java.lang.String getUserRefFromUserId(java.lang.String userId)

Translate the userId into a user entity reference, (may or may not validate the userId first)

Parameters:

userId - the internal user Id (not the eid or username)

Returns:

the user entity reference (e.g. /user/{userId})

### isEntityRequestInternal

boolean isEntityRequestInternal(java.lang.String reference)

Checks to see if the current entity request is internal (and therefore can optionally bypass some or all security), see the [RequestStorage](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestStorage.html) (from [RequestStorable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\RequestStorable.html) capability) for more info

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments (normally the id at least)

Returns:

true if the current request is internal OR false if external or REST based

### isUserAdmin

boolean isUserAdmin(java.lang.String userReference)

Check if this user has super admin level access (permissions)

Parameters:

userReference - the user entity reference (e.g. /user/{userId} - not id, eid, or username)

Returns:

true if the user has admin access, false otherwise

### isUserAllowedInEntityReference

boolean isUserAllowedInEntityReference(java.lang.String userReference,

java.lang.String permission,

java.lang.String reference)

Check if a user has a specified permission for the entity reference, primarily a convenience method for checking location permissions

Parameters:

userReference - the user entity reference (e.g. /user/{userId} - not id, eid, or username)

permission - a permission string constant

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments (normally the id at least)

Returns:

true if allowed, false otherwise

### populate

java.util.List<java.lang.String> populate(java.lang.Object object,

java.util.Map<java.lang.String,java.lang.Object> properties)

Populates an object with the values in the properties map, this will not fail if the fieldName in the map is not a property on the object or the fieldName cannot be written to with the value in the object. This will attempt to convert the provided object values into the right values to place in the object  
NOTE: simple types like numbers and strings can almost always be converted from just about anything though they will probably end up as 0 or ""  
Setting fields supports simple, nested, indexed, and mapped values:  
Simple: Get/set a field in a bean (or map), Example: "title", "id"  
Nested: Get/set a field in a bean which is contained in another bean, Example: "someBean.title", "someBean.id"  
Indexed: Get/set a list/array item by index in a bean, Example: "myList[1]", "anArray[2]"  
Mapped: Get/set a map entry by key in a bean, Example: "myMap(key)", "someMap(thing)"

Parameters:

object - any object

properties - a map of fieldNames -> Object

Returns:

the list of fieldNames which were successfully written to the object

Throws:

java.lang.IllegalArgumentException - if the arguments are invalid

### registerPermission

void registerPermission(java.lang.String permission)

Register a permission key as a valid permission for use in Sakai, permissions will not appear unless they are registered each time Sakai starts up so you should run this in your service init method

Parameters:

permission - the permission key (e.g.: toolname.read.all, toolname.delete.owned)

### restoreCurrentUser

java.lang.String restoreCurrentUser()

Restores the current user to the one from before [setCurrentUser(String)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\DeveloperHelperService.html#setCurrentUser%28java.lang.String%29) was called or does nothing if there was no previous user stored

Returns:

the restored current user reference OR null if there was no user to restore

### setCurrentUser

java.lang.String setCurrentUser(java.lang.String userReference)

Set the current user to match the supplied user reference, the current user reference will be stored and returned (may be null), this is primarily useful when you need to switch a user to an admin or to some other user temporarily OR there is no current user but something you are calling expects to find one

Parameters:

userReference - the user entity reference (e.g. /user/{userId} - not id, eid, or username)

Returns:

the previous current user entity reference

Throws:

java.lang.IllegalArgumentException - if the userReference is invalid

## Interface: EntityBroker

public interface EntityBroker

extends [PropertiesProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\PropertiesProvider.html), [TagProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\TagProvider.html), [SearchProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\SearchProvider.html)

## This service interface defines the capabilities of the entity broker system It allows Sakai system methods, developers, etc. to access Sakai entity information (new and old)

Methods:

### browseEntities

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> browseEntities(java.lang.String prefix,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search,

java.lang.String userReference,

java.lang.String associatedReference,

java.lang.String parentReference,

java.util.Map<java.lang.String,java.lang.Object> params)

Returns the list of entity information (and possibly entities) for a user to view while browsing an entity space, this is specially designed to support browsing and picking entities, not all entities support browsing  
If the entity type indicated by the prefix does not support browsing then this will return an empty list

Parameters:

prefix - the string which represents a type of entity handled by an entity provider, the entity prefix to search for browseable entities in, if the prefix does not support browsing then no entities will be returned

search - a search object which can define the order to return entities, search filters, and total number of entities returned, may be left empty

userReference - (optional) the unique entity reference for a user which is browsing the results, this may be null to indicate that only items which are visible to all users should be shown

associatedReference - (optional) a globally unique reference to an entity, this is the entity that the returned browseable data must be associated with (e.g. limited by reference to a location or associated entity), this may be null to indicate there is no association limit

parentReference - (optional) if not null then only the entities which have the referenced entity as a parent will be searched, if null then the parent/child relationship would be ignored when searching browseable entities

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

Returns:

a list of entity data objects which contain the reference, URL, display title and optionally other entity data

Throws:

java.lang.SecurityException - if the data cannot be accessed by the current user or is not publicly accessible

java.lang.IllegalArgumentException - if the prefix is invalid or the search is invalid

java.lang.IllegalStateException - if any other error occurs

### entityExists

boolean entityExists(java.lang.String reference)

Check if an entity exists by the globally unique reference string, (the global reference string will consist of the entity prefix and any local ID). If no [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) for the reference is found which implements [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html), this method will return true by default, in other words, this cannot determine if a legacy entity exists, only a new entity

Parameters:

reference - a globally unique reference to an entity (e.g. /myprefix/myid), consists of the entity prefix and optional segments (normally the id at least)

Returns:

true if the entity exists, false otherwise

Throws:

java.lang.IllegalArgumentException - if the reference is invalid

### executeCustomAction

[ActionReturn](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\ActionReturn.html) executeCustomAction(java.lang.String reference,

java.lang.String action,

java.util.Map<java.lang.String,java.lang.Object> params,

java.io.OutputStream outputStream)

This will execute a custom action for an entity or space/collection of entities  
This is meant for specialized usage as custom actions are typically meant to be executed by REST calls only

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments

action - key which will be used to trigger the action (e.g. promote, double, photo), can be triggered by a URL like so: /user/aaronz/promote

params - (optional) an optional set of params to pass along with this custom action request, typically used to provide information about the request, may be left null if not needed

outputStream - (optional) an OutputStream to place binary or long text data into, if this is used for binary data then the [ActionReturn](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\ActionReturn.html) should be returned with the correct encoding information and the output variable set to the OutputStream, may be left null if this custom action does not deal with binary streams

Returns:

an [ActionReturn](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\ActionReturn.html) which contains entity data or binary/string data OR null if there is no return for this action

Throws:

java.lang.UnsupportedOperationException - if there is no action with this key for this entity

java.lang.IllegalArgumentException - if there are required params that are missing or invalid

java.lang.IllegalStateException - if the action cannot be performed for some reason

### fetchEntities

java.util.List<?> fetchEntities(java.lang.String prefix,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search,

java.util.Map<java.lang.String,java.lang.Object> params)

Allows these entities to be fetched based on search parameters, this should never return null and if there are no entities then the list should be empty  
Note: The entity class types in the list need to be able to be resolved from the ClassLoader of the EntityBrokerManager (currently this means deployed into shared)  
  
These do not have to be model objects and may simply be something created (e.g. String, Map, etc.) to give to anyone calling this method

Parameters:

prefix - the string which represents a type of entity handled by an entity provider, if the prefix does not support fetching collections then no entities will be returned

search - a search object which can define the order to return entities, search filters, and total number of entities returned,  
NOTE: There are some predefined search keys which you may optionally use, provider are encourage to support the SEARCH\_\* search keys listed in this interface

Returns:

a list of entity objects (POJOs, Map, etc.) of the type handled by this provider OR empty if none found, will not return null

Throws:

java.lang.SecurityException - if the data cannot be accessed by the current user or is not publicly accessible

java.lang.IllegalArgumentException - if the reference is invalid or the search is invalid

java.lang.IllegalStateException - if any other error occurs

### fetchEntity

java.lang.Object fetchEntity(java.lang.String reference)

Fetches a concrete object representing this entity reference; either one from the [Resolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Resolvable.html) capability if implemented by the responsible [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), or else from the underlying legacy Sakai entity system  
Note that this may be a String or Map and does not have to be a POJO, the type of object should be determined out of band  
This will return null if the entity exists but is not [Resolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Resolvable.html) or available in the legacy entity system

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and local id

Returns:

an object which represents the entity OR null if none can be found or this type does not support fetching

Throws:

java.lang.SecurityException - if the entity cannot be accessed by the current user or is not publicly accessible

java.lang.IllegalArgumentException - if the reference is invalid

java.lang.IllegalStateException - if any other error occurs

### findEntitesByTags

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> findEntitesByTags(java.lang.String[] tags,

java.lang.String[] prefixes,

boolean matchAll,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search,

java.util.Map<java.lang.String,java.lang.Object> params)

Search for all entities which have the given tags, can limit the return using the search object

Parameters:

tags - a set of tags associated with entities

prefixes - (optional) a set of unique entity prefixes, limits the search to only include entities in these prefixes, if this is null then all entities and prefixes are searched  
NOTE: It is much more efficient to specify prefixes

matchAll - if true then all tags must exist on the entity for it to be matched, if false then the entity just has to have one or more of the given tags

search - (optional) a search object, used to order or limit the number of returned results, restrictions will be typically ignored

params - (optional) an optional set of params to pass along with this custom action request, typically used to provide information about the request, may be left null if not needed

Returns:

a list of entity search results (contains the ref, url, displayname of the matching entities)

Throws:

java.lang.IllegalArgumentException - if the tags set is empty or null

### fireEntityRequest

[EntityResponse](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\EntityResponse.html) fireEntityRequest(java.lang.String reference,

java.lang.String viewKey,

java.lang.String format,

java.util.Map<java.lang.String,java.lang.String> params,

java.lang.Object entity)

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optionally the local id, cannot be null or empty

viewKey - specifies what kind of request this is (create, read/show, etc.), must correspond to the VIEW constants in [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html), example: [EntityView.VIEW\_SHOW](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#VIEW_SHOW)

format - (optional) this is the format for this request (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html), e.g. XML), if nothing is specified then the default will be used: [Formats.HTML](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html#HTML)

params - (optional) any params you want to send along with the request should be included here, they will be placed into the query string or the request body depending on the type of request this is

entity - (optional) leave this null in most cases, if you supply an entity object here it will be encoded based on the supplied format (only if the entity supports output formatting) and then decoded on the other end (only if the entity supports input translation), in most cases it is better to supply the entity values in the params

Returns:

the response information encoded in an object, you must check this to see what the results of the request were (getting a response back does not mean the request succeeded)

Throws:

java.lang.IllegalArgumentException - if the inputs are invalid

java.lang.RuntimeException - if the http request has an unrecoverable failure or an encoding failure occurs

### fireEvent

void fireEvent(java.lang.String eventName,

java.lang.String reference)

Fire an event to Sakai with the specified name, targeted at the supplied reference, which should be a reference to an existing entity managed by this broker  
NOTE: This will allow events to be fired for references without a broker or invalid references

Parameters:

eventName - a string which represents the name of the event (e.g. announcement.create), cannot be null or empty

reference - a globally unique reference to an entity, consists of the entity prefix and optionally the local id, cannot be null or empty

### formatAndOutputEntity

void formatAndOutputEntity(java.lang.String reference,

java.lang.String format,

java.util.List<?> entities,

java.io.OutputStream output,

java.util.Map<java.lang.String,java.lang.Object> params)

Format and output an entity or collection included or referred to by this entity ref object into output according to the format string provided, Should take into account the reference when determining what the entities are and how to encode them

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments

format - a string constant indicating the format (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html)) for output, (example: #XML)

entities - (optional) a list of entities or [EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html) objects to create formatted output for, if this is null then the entities will be retrieved based on the reference, if this contains only a single item AND the ref refers to a single entity then the entity will be extracted from the list and encoded without the indication that it is a collection, for all other cases the encoding will include an indication that this is a list of entities

output - the output stream to place the formatted data in, should be UTF-8 encoded if there is char data

params - (optional) an optional set of params to pass along with this custom action request, typically used to provide information about the request, may be left null if not needed

Throws:

[FormatUnsupportedException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\FormatUnsupportedException.html) - if entity cannot handle this format type

java.lang.IllegalArgumentException - if any of the arguments are invalid

[EntityEncodingException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityEncodingException.html) - is there is failure encoding the output

java.lang.IllegalStateException - for all other failures

### getBrowseableEntities

java.util.List<[BrowseEntity](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\BrowseEntity.html)> getBrowseableEntities(java.lang.String parentPrefix)

For authors of entity browsing systems, this provides a list of all the meta data related to the entities in the system which are [Browseable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Browseable.html), this provides the root entities when there is no parent prefix provided

Parameters:

parentPrefix - (optional) the parent prefix to get the browseable entities for, if this is null then all the root browseable entities are returned

Returns:

the list of meta data for all entity prefixes that are [Browseable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Browseable.html)

### getEntities

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> getEntities(java.lang.String prefix,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search,

java.util.Map<java.lang.String,java.lang.Object> params)

Gets entity data (and possibly entities) for a specific entity prefix, entity data contains the reference, URL, display title and optionally the concrete entity and properties  
If the entity type indicated by the prefix does not support collections then this will return an empty list

Parameters:

prefix - the string which represents a type of entity handled by an entity provider, the entity prefix to search for the collection of entities in, if the prefix does not support browsing then no entities will be returned

search - a search object which can define the order to return entities, search filters, and total number of entities returned, may be left empty

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

Returns:

a list of entity data objects OR an empty list if none found

Throws:

java.lang.SecurityException - if the data cannot be accessed by the current user or is not publicly accessible

java.lang.IllegalArgumentException - if the prefix is invalid or the search is invalid

java.lang.IllegalStateException - if any other error occurs

### getEntity

[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html) getEntity(java.lang.String reference)

Gets the data related to an entity as long as it exists, always includes at least the entity reference information and the URL, may also include a concrete entity object and entity properties

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and local id

Returns:

an entity data object which contains data about the entity OR null if the entity does not exist

Throws:

java.lang.SecurityException - if the entity cannot be accessed by the current user or is not publicly accessible

java.lang.IllegalArgumentException - if the reference is invalid

java.lang.IllegalStateException - if any other error occurs

### getEntityURL

java.lang.String getEntityURL(java.lang.String reference)

Get the full absolute URL to the entity defined by this entity reference, this will fail-safe to a direct URL to an entity space URL if that is all that is available

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments

Returns:

a full URL string (e.g. http://server/direct/prefix/id)

Throws:

java.lang.IllegalArgumentException - if the reference or other inputs are invalid

### getEntityURL

java.lang.String getEntityURL(java.lang.String reference,

java.lang.String viewKey,

java.lang.String extension)

Get the full absolute URL to the entity view defined by these params, this will fail-safe to a direct URL to an entity space URL if that is all that is available, this will use the default entity URL template associated with the viewKey and include an optional extension if specified (these will be inferred if they are missing)

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optionally the local id

viewKey - the specific view type to get the URL for, use the VIEW\_\* constants from [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) (e.g. [EntityView.VIEW\_LIST](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#VIEW_LIST)), can be null to determine the key automatically

extension - the optional extension to add to the end which defines the expected data which is returned, use constants in [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html) (e.g. [Formats.XML](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html#XML)), can be null to use no extension, default is assumed to be html if none is set

Returns:

the full URL string to a specific entity or space, (e.g. http://server/direct/prefix/id)

Throws:

java.lang.IllegalArgumentException - if the reference or other inputs are invalid

### getEntityView

[EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) getEntityView(java.lang.String reference,

java.lang.String viewKey,

java.lang.String extension)

Get the [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) object which represents a specific view of an entity or entity collection, this is similar to [getEntityURL(String, String, String)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityBroker.html#getEntityURL%28java.lang.String,%20java.lang.String,%20java.lang.String%29) but allows the developer to deal with the [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) object if desired

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optionally the local id

viewKey - the specific view type to get the URL for, use the VIEW\_\* constants from [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) (e.g. [EntityView.VIEW\_LIST](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#VIEW_LIST)), can be null to determine the key automatically

extension - the optional extension to add to the end which defines the expected data which is returned, use constants in [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html) (e.g. [Formats.XML](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html#XML)), can be null to use no extension, default is assumed to be html if none is set

Returns:

an EntityView object if one can be formed

Throws:

java.lang.IllegalArgumentException - if the params cannot be made into an EntityView

### getRegisteredPrefixes

java.util.Set<java.lang.String> getRegisteredPrefixes()

Retrieve a complete set of all currently registered [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) prefixes

Returns:

all currently registered entity prefixes

### isPrefixRegistered

boolean isPrefixRegistered(java.lang.String prefix)

Check if a prefix is currently registered

Parameters:

prefix - the string which represents a type of entity handled by an entity provider

Returns:

true if the prefix is registered OR false if not

### parseReference

[EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) parseReference(java.lang.String reference)

Parses an entity reference into a concrete object, of type [EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html), or some class derived from it, for example IdEntityReference or some other class of object which is returned from ParseSpecParseable.

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments

Returns:

an entity reference object which will contain the entity prefix and any optional segments, or null if the reference was not recognized as a valid entity handled by the broker (will be an IdEntityReference if there is an id set)

Throws:

java.lang.IllegalArgumentException - if the reference is invalid and cannot be parsed

### translateInputToEntity

java.lang.Object translateInputToEntity(java.lang.String reference,

java.lang.String format,

java.io.InputStream input,

java.util.Map<java.lang.String,java.lang.Object> params)

Translates the input data stream in the supplied format into an entity object for this reference

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optional segments

format - a string constant indicating the format (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html)) of the input, (example: #XML)

input - a stream which contains the data to make up this entity, you may assume this is UTF-8 encoded if you don't know anything else about it

params - (optional) set of parameters which may be used to control this request, may be left null if not needed

Returns:

an entity object of the type used for the given reference

Throws:

[FormatUnsupportedException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\FormatUnsupportedException.html) - if entity cannot handle this format type

java.lang.IllegalArgumentException - if any of the arguments are invalid

[EntityEncodingException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityEncodingException.html) - is there is failure translating the input

java.lang.IllegalStateException - for all other failures.

## Interface: EntityBrokerManager

public interface EntityBrokerManager

The core of the EB system, this is generally not for use by developers and is mostly for internal use but it should be ok to use most of these methods

Methods:

### browseEntities

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> browseEntities(java.lang.String prefix,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search,

java.lang.String userReference,

java.lang.String associatedReference,

[EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) parentEntityRef,

java.util.Map<java.lang.String,java.lang.Object> params)

Fetches the browseable entities

Parameters:

prefix -

search -

userReference -

associatedReference -

parentEntityRef -

params -

Returns:

a list of entity data results to browse

### convertToEntityData

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> convertToEntityData(java.util.List<?> entities,

[EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref)

Convert a list of objects to entity data objects (also populates them), will preserve null (i.e. null in => null out)

### convertToEntityData

[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html) convertToEntityData(java.lang.Object entity,

[EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref)

Convert a single object to an entity data object (also populates it), will preserve null (i.e. null in => null out)

### entityExists

boolean entityExists([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref)

Determines if an entity exists based on the reference

Parameters:

reference - an entity reference object

Returns:

true if entity exists, false otherwise

### fetchEntities

java.util.List<?> fetchEntities([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search,

java.util.Map<java.lang.String,java.lang.Object> params)

Get a list of entities from [CollectionResolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\CollectionResolvable.html) first if available or [BrowseSearchable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\BrowseSearchable.html) if not, returns the entities as actual entities (converts from [EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html) if that was used), correctly handles references to single entities as well

Parameters:

ref - the reference

search - a search (should not be null)

params -

Returns:

a list of entities OR empty list if none found for the given reference

### fetchEntity

java.lang.Object fetchEntity([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref)

Get an entity object of some kind for this reference if it has an id, will simply return null if no id is available in this reference

Parameters:

ref - an entity reference

Returns:

the entity object for this reference OR null if none can be retrieved

### getBrowseableEntities

java.util.List<[BrowseEntity](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\BrowseEntity.html)> getBrowseableEntities(java.lang.String parentPrefix)

Get the meta data about browseable entities

Parameters:

parentPrefix - the prefix of the parent type (null for the root types)

Returns:

the list of browseable entity meta data

### getEntitiesData

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> getEntitiesData([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search,

java.util.Map<java.lang.String,java.lang.Object> params)

Get a list of entities from [CollectionResolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\CollectionResolvable.html) first if available or [BrowseSearchable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\BrowseSearchable.html) if not, returns the entities wrapped in [EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html), correctly handles references to single entities as well

Parameters:

ref - the reference

search - a search (should not be null)

params -

Returns:

a list of entities OR empty list if none found for the given reference

### getEntityData

[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html) getEntityData([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref)

Get the entity data for a reference if possible

Parameters:

ref - an entity reference

Returns:

an [EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html) object for this reference if one can be found OR null if not

### getEntityPropertiesService

[EntityPropertiesService](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\providers\EntityPropertiesService.html) getEntityPropertiesService()

Allows access to the current EntityPropertiesService

Returns:

the current EntityPropertiesService

### getEntityProviderManager

[EntityProviderManager](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProviderManager.html) getEntityProviderManager()

Allows access to the current EntityProviderManager service

Returns:

the current EntityProviderManager service

### getEntityProviderMethodStore

[EntityProviderMethodStore](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProviderMethodStore.html) getEntityProviderMethodStore()

Allows access to the current EntityProviderMethodStore service

Returns:

the current EntityProviderMethodStore

### getEntityRESTProvider

[EntityRESTProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\providers\EntityRESTProvider.html) getEntityRESTProvider()

Allows access to the registered REST provider if there is one

Returns:

the REST provider OR null if there is not one

### getEntityURL

java.lang.String getEntityURL(java.lang.String reference,

java.lang.String viewKey,

java.lang.String extension)

Creates the full URL to an entity using the sakai ServerConfigurationService, (e.g. http://server:8080/direct/entity/123/)  
  
Note: the webapp name (relative URL path) of the direct servlet, of "/direct" is hardcoded into this method, and the org.sakaiproject.entitybroker.util.servlet.DirectServlet must be deployed there on this server.

Parameters:

reference - a globally unique reference to an entity, consists of the entity prefix and optionally the local id

viewKey - the specific view type to get the URL for, can be null to determine the key automatically

extension - the optional extension to add to the end, can be null to use no extension

Returns:

the full URL to a specific entity or space

### getEntityViewAccessProviderManager

[EntityViewAccessProviderManager](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProviderManager.html) getEntityViewAccessProviderManager()

Allows access to the current EntityViewAccessProviderManager service

Returns:

the current EntityViewAccessProviderManager

### getExternalIntegrationProvider

[ExternalIntegrationProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\providers\ExternalIntegrationProvider.html) getExternalIntegrationProvider()

Allows access to any registered ExternalIntegrationProvider

Returns:

the external integration provider OR null if there is not one

### getMaxJSONLevel

int getMaxJSONLevel()

Returns the maximum depth of object graph allowed during transcoding to JSON

### getRequestGetter

[RequestGetterWrite](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestGetterWrite.html) getRequestGetter()

Allows access to the current RequestGetter service

Returns:

the current RequestGetter

### getRequestStorage

[RequestStorageWrite](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestStorageWrite.html) getRequestStorage()

Allows access to the current RequestStorageWrite service

Returns:

the current RequestStorageWrite

### getSampleEntityObject

java.lang.Object getSampleEntityObject(java.lang.String prefix,

java.lang.String id)

Safely get the sample entity object which is defined for a prefix, if there is not one then return null

Parameters:

prefix - the entity prefix

id - (optional) will get the actual entity for this id as a sample

Returns:

a sample object OR null if none can be found

### getServletContext

java.lang.String getServletContext()

FOR INTERNAL USE ONLY (do not mess with this in other words) This gets the known REST servlet context if there is one, will return a default "/rest" if none is known, will not return "" or null

### getVersionInfo

java.lang.String getVersionInfo()

Used to get a version info string which can be output to see what version we are working with

### makeEntityView

[EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) makeEntityView([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref,

java.lang.String viewKey,

java.lang.String extension)

Reduce code duplication and ensure custom templates are used

### makeFullURL

java.lang.String makeFullURL(java.lang.String pathURL)

Make a full entity URL (http://....) from just a path URL (/prefix/id.xml)

Parameters:

pathURL - a path (like pathInfo from a request) (e.g. /prefix/id.xml)

Throws:

java.lang.IllegalArgumentException - is the pathURL is null

### parseEntityURL

[EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) parseEntityURL(java.lang.String entityURL)

Parses an entity URL into an entity view object, handles custom parsing templates

Parameters:

entityURL - an entity URL

Returns:

the entity view object representing this URL or null if there is no provider found for the prefix parsed out

Throws:

java.lang.IllegalArgumentException - if there is a failure during parsing

### parseReference

[EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) parseReference(java.lang.String reference)

Parses an entity reference into the appropriate reference form

Parameters:

reference - a unique entity reference

Returns:

the entity reference object or null if there is no provider found for the prefix parsed out

Throws:

java.lang.IllegalArgumentException - if there is a failure during parsing

### populateEntityData

void populateEntityData([EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)[] data)

Add in the extra meta data (URL, title, etc.) to all entity data objects, handles it as efficiently as possible without remaking an entity view on every call, this is fail safe (i.e. it should throw no exceptions)

Parameters:

data - a list of entity data

### populateEntityData

void populateEntityData(java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> data)

Add in the extra meta data (URL, title, etc.) to all entity data objects, handles it as efficiently as possible without remaking an entity view on every call, this is fail safe (i.e. it should throw no exceptions)

Parameters:

data - a list of entity data

### setEntityRESTProvider

void setEntityRESTProvider([EntityRESTProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\providers\EntityRESTProvider.html) entityRESTProvider)

Allows the developer to set a REST provider to add functionality to the [EntityBroker](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityBroker.html) system from a REST handler, the system will operate without this set but some methods will fail

Parameters:

entityRESTProvider - a service to provide REST functionality

### setExternalIntegrationProvider

void setExternalIntegrationProvider([ExternalIntegrationProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\providers\ExternalIntegrationProvider.html) externalIntegrationProvider)

Allows developers to setup providers to handle parts of the EB system which cannot really be handled internally, the system will operate without this set

Parameters:

externalIntegrationProvider - the external integration provider to use in the system

### setServletContext

void setServletContext(java.lang.String servletContext)

FOR INTERNAL USE ONLY (do not mess with this in other words)

Parameters:

servletContext - sets the servlet context being used by the system (defaults to #DIRECT)

## Class: EntityReference

public class EntityReference implements java.lang.Cloneable, java.io.Serializable

The class of all Entity references handled by the EntityBroker system. This provides the entity prefix, which uniquely identifies the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) responsible for handling the Entity. It optionally provides the entity id which uniquely identifies an entity locally within the prefix space. It also provides for all parsing methods for entity references (always of the form /prefix or /prefix/id)

Fields:

### PERIOD

public static final transient char PERIOD

### prefix

public java.lang.String prefix

An entity prefix, should match with the prefix handled in an [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), uniquely identifies an entity space or entity type  
WARNING: use the [getPrefix()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html#getPrefix%28%29) method rather than referring to this directly

### SEPARATOR

public static final transient char SEPARATOR

### serialVersionUID

public static final long serialVersionUID

Methods:

### copy

public [EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) copy()

Returns:

a copy of this reference

### copy

public static [EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) copy([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref)

Creates a copy of an entity reference

Parameters:

ref - an entity reference object

Returns:

the new copy of the ref

Throws:

java.lang.IllegalArgumentException - if the ref is invalid OR null

### getId

public java.lang.String getId()

A local entity id, represents an entity uniquely in a tool/webapp, could match with the actual id of a model data object, this will be null if this reference refers to an entity space only

Returns:

the entity id (locally unique id for an entity of this entity type) or null if this this reference refers to an entity space only

### getIdFromRef

public static java.lang.String getIdFromRef(java.lang.String reference)

Will convert a reference into an id (even if it is not a reference)

Parameters:

reference - an entity reference (e.g. /user/aaronz)

Returns:

the id from the reference (e.g. aaronz), preserves null

### getIdFromRefByKey

public static java.lang.String getIdFromRefByKey(java.lang.String reference,

java.lang.String key)

Get the id value out of a reference by the key that preceeds it

Parameters:

reference - an entity reference (e.g. /site/siteId/group/groupId)

key - the key to get the id from (e.g. 'group' yields groupId)

Returns:

the id value OR null if no key is found or no id is available

### getOriginalReference

public java.lang.String getOriginalReference()

This is a special method and should not normally be used, use [toString()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html#toString%28%29) or [getReference()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html#getReference%28%29)

Returns:

the reference string used to create this object or null if this was created using a constructor which does not supply a reference

### getPrefix

public java.lang.String getPrefix()

Returns:

the entity prefix (uniquely identifies an entity space or entity type), this should never be null

### getPrefix

public static java.lang.String getPrefix(java.lang.String reference)

Get the entity prefix based on an entity reference string, WARNING: this is meant for internal use, use [EntityReference(String)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html#EntityReference%28java.lang.String%29) and the methods in [EntityBroker](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityBroker.html) to parse references

Parameters:

reference - an entity reference or entity URL

Returns:

the entity prefix

### getReference

public java.lang.String getReference()

Get the string reference for this entity reference object, same as calling [toString()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html#toString%28%29)

Returns:

the full entity reference

### getSpaceReference

public java.lang.String getSpaceReference()

Get the space reference for this entity reference, this ignored any id and only returns the reference to the entity space

Returns:

the entity space reference (e.g. /myPrefix)

### isEmpty

public boolean isEmpty()

Returns:

true if this object contains no reference data, false if it contains a valid reference

### toString

public java.lang.String toString()

Overrides:

toString in class java.lang.Object

Returns:

the string version of this entity reference, example: /prefix if there is no id or /prefix/id if there is an id

## Class: EntityView

public class EntityView implements java.lang.Cloneable, java.io.Serializable

Defines an entity view (a specific way to looking at entity data, e.g. LIST of entities, SHOW a single entity, UPDATE an entity, DELETE an entity, create a NEW entity), each view has a unique view key constant related to it (e.g. [VIEW\_LIST](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#VIEW_LIST))   
The view contains all the known information about a view request including the entity prefix, reference, full path and segments, format (extension), method (POST, GET, etc.), and view key (type). The Entity View can generate the URL for this view based on the data it contains.   
Views use URL templates which can be controlled via custom templates if desired.

## Methods:

### clone

public java.lang.Object clone()

throws java.lang.CloneNotSupportedException

Overrides:

clone in class java.lang.Object

Throws:

java.lang.CloneNotSupportedException

### copy

public [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) copy()

Returns:

a copy of this object

### copy

public static [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) copy([EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) ev)

Makes a copy of an EntityView which can be changed independently

Parameters:

ev - any EntityView

Returns:

the copy

Throws:

java.lang.IllegalArgumentException - if the input is null OR not completely constructed

### getAnazlyzedTemplates

public java.util.List<[TemplateParseUtil.PreProcessedTemplate](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.PreProcessedTemplate.html)> getAnazlyzedTemplates()

### getEntityReference

public [EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) getEntityReference()

Returns:

the entity reference object which indicates which entity this view related to

### getEntityURL

public java.lang.String getEntityURL()

Returns:

the entity URL of the internal reference based on the internal viewKey and extension, defaults to [TemplateParseUtil.TEMPLATE\_SHOW](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#TEMPLATE_SHOW) or the [TemplateParseUtil.TEMPLATE\_LIST](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#TEMPLATE_LIST) one if there is no id, example: /prefix if there is no id or /prefix/id if there is an id

Throws:

java.lang.IllegalArgumentException - if there is not enough information to generate a URL

### getEntityURL

public java.lang.String getEntityURL(java.lang.String viewKey,

java.lang.String extension)

Get an entityUrl by merging a specific template with the data in this EB object

Parameters:

viewKey - a key which uniquely identifies a view, from the set of template keys #PARSE\_TEMPLATE\_KEYS

extension - an optional extension related to this view (e.g. xml), do not include the period, leave this null for no extension

Returns:

the entityUrl which goes to this view

Throws:

java.lang.IllegalArgumentException - if the viewKey is invalid OR there is not enough information in the path segments to generate the requested URL

### getExtension

public java.lang.String getExtension()

The extension for this view which defines the type of data that will be returned for this view, examples: html, xml, json NOTE: you should assume html return format when this is null

Returns:

the extension for this view if there is one, this will be null if there was no extension in the original entityUrl

### getFormat

public java.lang.String getFormat()

Returns:

the format (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html)) that is being used for this view, will return [Formats.HTML](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html#HTML) if none set

### getMethod

public java.lang.String getMethod()

Returns:

the method (GET, POST, etc.) (from [EntityView.Method](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.Method.html)) being used for this view, defaults to GET if none was set explicitly

### getOriginalEntityUrl

public java.lang.String getOriginalEntityUrl()

Special use only, normally you should use [toString()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#toString%28%29) or [getEntityURL(String, String)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#getEntityURL%28java.lang.String,%20java.lang.String%29)

Returns:

the original entity URL which was used to create this entity view, includes the optional pieces from the URL, will be null if this was created without using a constructor that takes an entityUrl

### getParseTemplate

public java.lang.String getParseTemplate(java.lang.String templateKey)

Parameters:

templateKey - a key from the set of template keys #PARSE\_TEMPLATE\_KEYS, should match with the viewKey

Returns:

the template being used by this entity view for this key or null if none found

### getParseTemplates

public java.util.List<[TemplateParseUtil.Template](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.Template.html)> getParseTemplates()

### getPathSegment

public java.lang.String getPathSegment(int position)

Get a segment value by position from the encoded URL for this view  
Position 0 is always the prefix   
Example: /user/aaronz/promote/stuff.xml   
position 0: 'user'   
position 1: 'aaronz'   
position 2: 'promote'   
position 3: 'stuff'   
position 4: null

Parameters:

position - the position number in the path segments, 0 is always the prefix

Returns:

the value at the given path position OR null if there is nothing at that position

### getPathSegment

public java.lang.String getPathSegment(java.lang.String parseVariable)

Gets the parsed values of path segment variables

Parameters:

parseVariable - a path segment variable (the thing that gets replaced in the parse template)

Returns:

the value of this parse variable or null if no value exists

### getPathSegments

public java.lang.String[] getPathSegments()

Get all the path segments for the encoded URL for this view  
Example: /user/aaronz/promote/stuff.xml   
segments = {"user","aaronz","promote","stuff"}

Returns:

an array of path segments

### getViewKey

public java.lang.String getViewKey()

Returns:

the key which uniquely identifies the view we are associated with, uses the constants like [VIEW\_LIST](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#VIEW_LIST) and [VIEW\_NEW](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#VIEW_NEW)

### loadParseTemplates

public void loadParseTemplates(java.util.List<[TemplateParseUtil.Template](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.Template.html)> templates)

Override this method if creating a custom [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) object

Parameters:

templates - a list of template constants -> parse templates, the array which defines the set of template keys is #PARSE\_TEMPLATE\_KEYS  
Rules for parse templates: 1) "{","}", and [SEPARATOR](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#SEPARATOR) are special characters and must be used as indicated only 2) Must begin with a [SEPARATOR](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#SEPARATOR), must not end with a [SEPARATOR](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#SEPARATOR) 3) must begin with "/{prefix}" (use the [SEPARATOR](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#SEPARATOR) and [PREFIX](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#PREFIX) constants) 3) each {var} can only be used once in a template 4) {var} can never touch each other (i.e /{var1}{var2}/{id} is invalid) 5) each {var} can only have the chars from [TemplateParseUtil.VALID\_VAR\_CHARS](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#VALID_VAR_CHARS) 6) parse templates can only have the chars from [TemplateParseUtil.VALID\_TEMPLATE\_CHARS](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#VALID_TEMPLATE_CHARS) 7) Empty braces ({}) cannot appear in the template

### parseEntityURL

public void parseEntityURL(java.lang.String entityURL)

Used to build this object after it has already been created (typically so custom templates can be inserted)

Parameters:

entityURL - a URL path which goes to a specific entity view, consists of path segments defined by path templates and includes an option extension

### preloadParseTemplates

public void preloadParseTemplates(java.util.List<[TemplateParseUtil.PreProcessedTemplate](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.PreProcessedTemplate.html)> preprocessedTemplates)

Special efficiency method to reduce reloading of custom templates, do not use this unless you wrote it or REALLY know what you are doing

### setEntityReference

public [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) setEntityReference([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref)

Allows for easy chained construction of EntityViews by setting an EntityReference, does not set the viewkey or extension unless they are unset, maintains current extension

### setExtension

public void setExtension(java.lang.String extension)

### setMethod

public void setMethod([EntityView.Method](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.Method.html) method)

### setViewKey

public void setViewKey(java.lang.String viewKey)

### toString

public java.lang.String toString()

Overrides:

toString in class java.lang.Object

Returns:

the entity URL of the internal reference based on the internal viewKey and extension, defaults to [TemplateParseUtil.TEMPLATE\_SHOW](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#TEMPLATE_SHOW) or the [TemplateParseUtil.TEMPLATE\_LIST](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#TEMPLATE_LIST) one if there is no id, example: /prefix if there is no id or /prefix/id if there is an id

Throws:

java.lang.IllegalArgumentException - if there is not enough information to generate a URL

### translateViewKeyToMethod

public static [EntityView.Method](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.Method.html) translateViewKeyToMethod(java.lang.String viewKey)

Translate a viewkey into an http method

Parameters:

viewKey -

Returns:

the method which matches this viewkey

Test Classes:

## Package: org.sakaiproject.entitybroker.access

## Interface: AccessFormats

public interface AccessFormats

extends [EntityViewAccessProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProvider.html), [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html)

Indicates that entity requests can be handled for certain formats which are indicated

Methods:

### getHandledAccessFormats

java.lang.String[] getHandledAccessFormats()

Defines the access format types (extensions) handled by this access provider  
The default if this interface is not implemented is to pass through all requests to the access provider that is defined

Returns:

an array containing the format types (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html)) handled   
OR empty array to indicate all are handled (same as not implementing [AccessFormats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\AccessFormats.html))   
OR null to indicate none are handled (same as not implementing [EntityViewAccessProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProvider.html))

## Interface: AccessViews

public interface AccessViews

extends [EntityViewAccessProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProvider.html)

Indicates that entity requests can be handled for certain views which are indicated

Methods:

### getHandledEntityViews

java.lang.String[] getHandledEntityViews()

Defines the entity views handled by this access provider  
The default if this interface is not implemented is to pass through all requests to the access provider that is defined

Returns:

an array containing the entity view keys (from [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html)) handled   
OR empty array to indicate all are handled (same as not implementing [AccessViews](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\AccessViews.html))   
OR null to indicate none are handled (same as not implementing [EntityViewAccessProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProvider.html))

## Interface: EntityViewAccessProvider

public interface EntityViewAccessProvider

Represents a bean which is capable of handling access for an [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html), this replaces the HttpServletAccessProvider as all entity URLs are now being parsed so more information can be provided through the [EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html)  
  
This interface would be typically implemented from a tool (webapp) context, and registered with the [EntityViewAccessProviderManager](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProviderManager.html) in a context loader listener  
  
If the implementation throws a SecurityException during the course of this method, the access will be directed to a login page or authentication method before being redirected back to the implementation method  
If you want to control the requests which make it through to this by format type you can optionally implement [AccessFormats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\AccessFormats.html)

## Methods:

### handleAccess

void handleAccess([EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) view,

javax.servlet.http.HttpServletRequest req,

javax.servlet.http.HttpServletResponse res)

Make and return the data responses for this type of data provider for a specific entity view and entity reference (contained within the entity view), use the request to get any additional sent in information you may need or want and use the response to hold the output you generate  
  
NOTE: If you decide that you cannot handle this access request for any reason you can either throw an [EntityException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityException.html) to specify why OR throw a general Exception, both will kill the request entirely but the general exception will pass through the system while the [EntityException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityException.html) will produce a handled result

Parameters:

view - an entity view, should contain all the information related to the incoming entity URL

req - the servlet request (available in case you need to get anything out of it)

res - the servlet response, put the correct data response into the outputstream

Throws:

[FormatUnsupportedException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\FormatUnsupportedException.html) - if the format requested in the view is not supported

[EntityException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityException.html) - if there is a request processing/handling failure

## Interface: EntityViewAccessProviderManager

public interface EntityViewAccessProviderManager

Manages all the access providers for the entity views in the system  
Use this to register yourself as handling the entity views for a set of entites (based on the prefix) This will be used by the entity broker to determine if anyone is handling entity views for an entity type

Methods:

### getProvider

[EntityViewAccessProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProvider.html) getProvider(java.lang.String prefix)

Get an entity view access provider for a prefix if one exists

Parameters:

prefix - the unique entity prefix that defines the entity type handled

Returns:

the provider related to this prefix or null if no provider can be found

### registerProvider

void registerProvider(java.lang.String prefix,

[EntityViewAccessProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProvider.html) provider)

Register a provider as handling entity view requests and delivering responses

Parameters:

prefix - the unique entity prefix that defines the entity type handled by this provider

provider - the actual provider bean

### unregisterProvider

void unregisterProvider(java.lang.String prefix)

Removes the provider from the registered set for this prefix

Parameters:

prefix - the unique entity prefix that defines the entity type handled by this provider

## Package: org.sakaiproject.entitybroker.collector

## Interface: AutoRegister

public interface AutoRegister

By implementing this interface you are telling the system to pick up all the objects which are implementing this interface and put them in a bucket so they will be loaded into any bean which implements BeanCollector and asks for them

## Interface: BeanCollector<T>

public interface BeanCollector<T>

Implement this in order to cause spring to inject a set of beans into your spring bean (if any exist) which implement the interface you define (they will also have to implement [AutoRegister](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\collector\AutoRegister.html))

Methods:

### getCollectedType

java.lang.Class<[T](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\collector\BeanCollector.html)> getCollectedType()

This allows the developer to set the type of the beans which they want collected

Returns:

the class type of the beans to collect

### setCollectedBeans

void setCollectedBeans(java.util.List<[T](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\collector\BeanCollector.html)> collectedBeans)

This setter will be called automatically and the beans which are being collected for you will be placed in the List

Parameters:

collectedBeans - a list of all collected beans which were autoregistered

## Interface: BeanMapCollector

public interface BeanMapCollector

Implement this in order to cause spring to inject a map of type -> a set of beans into your spring bean (if any exist) which implement the type interfaces you define (those interfaces will also have to implement [AutoRegister](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\collector\AutoRegister.html))

Methods:

### getCollectedTypes

java.lang.Class<?>[] getCollectedTypes()

This allows the developer to set the types of the beans which they want collected

Returns:

the class type of the beans to collect

### setCollectedBeansMap

void setCollectedBeansMap(java.util.Map<java.lang.Class<?>,java.util.List<?>> collectedBeans)

This setter will be called automatically and the beans which are being collected for you will be placed in the map such that it is a map of class type -> list of beans of that type

Parameters:

collectedBeans - a list of all collected beans which were autoregistered

## Interface: OrderedBean

public interface OrderedBean

extends [AutoRegister](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\collector\AutoRegister.html)

This sets the order of a bean when it is grabbed by the bean collector and ensures a guaranteed priority order of collected beans, lower orders (numbers) will be loaded first and the orders do not have to be consecutive (there can be gaps)

Methods:

### getOrder

int getOrder()

Sets the order to load the bean which implements this method compared to other beans of the same type, lower orders (numbers) will be loaded first (i.e. order 1 will appear before order 3 in the list) and the orders do not have to be consecutive (there can be gaps), 2 beans with the same order or beans with no order set will be ordered randomly

Returns:

an int which represents the loading order

## Package: org.sakaiproject.entitybroker.entityprovider

## Interface: CoreEntityProvider

public interface CoreEntityProvider

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

This is the base unit for working with Sakai entities, by implementing this interface and creating a spring bean you will tie your entities into Sakai, there are many other interfaces which you can implement to extend the interaction of your entities with Sakai in this package  
You (the implementor) will want to create one implementation of this interface for each type of entity you want to link to Sakai to track events, provide URL access, etc.  
  
Usage:  
1) Implement this interface  
2) Implement any additional capabilities interfaces (optional)  
3) Create a spring bean definition in the Sakai application context (components.xml)  
4) Implement [AutoRegisterEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\AutoRegisterEntityProvider.html) or register this implementation some other way  
  
Recommended best practices: (example: Thing entity)  
1) Create an interface called ThingEntityProvider which extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) in api logic (add an entity package for it), (e.g. org.sakaiproject.evaluation.logic.entity.EvaluationEntityProvider.java)   
2) Add a public static string which contains the entity prefix (called ENTITY\_PREFIX), (e.g. public final static String ENTITY\_PREFIX = "eval-evaluation";)   
3) Implement your ThingEntityProvider in impl logic as ThingEntityProviderImpl (add an entity package for it), (e.g. org.sakaiproject.evaluation.logic.impl.entity.EvaluationEntityProviderImpl.java)   
4) Implement [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html) in ThingEntityProviderImpl   
5) Implement [AutoRegisterEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\AutoRegisterEntityProvider.html) in ThingEntityProviderImpl   
6) Add a spring bean definition in the Sakai application context (components.xml), use the api name as the id  
Example:

<bean id="org.sakaiproject.evaluation.logic.entity.EvaluationEntityProvider"

class="org.sakaiproject.evaluation.logic.impl.entity.EvaluationEntityProviderImpl"> </bean>

7) Add the needed maven dependendencies to api/logic and impl/logic project.xml files  
Exmaple:

<dependency>

<groupId>sakaiproject</groupId>

<artifactId>sakai-entitybroker-api</artifactId>

<version>${sakai.version}</version>

</dependency>

That should do it. You should now be able to use the [EntityBroker](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityBroker.html) to access information about your entities and register events for your entities (among other things).

## Methods:

### entityExists

boolean entityExists(java.lang.String id)

Check if a specific entity managed by this provider exists.  
This is primarily used to validate references before making other calls or operating on them.  
WARNING: This will be called many times and AT LEAST right before calls are made to any methods or capabilities related to specific entities, please make sure this is very efficient. If you are concerned about efficiency, it is ok for this method to always return true but you will no longer be able to be sure that calls through to your capability implementations are always valid.

Parameters:

id - a locally unique id for an entity managed by this provider  
NOTE: this will be an empty string if this is an entity space (singleton entity) without an id available

Returns:

true if an entity with given local id exists, false otherwise

## Interface: EntityProvider

public interface EntityProvider

Beans may implement this interface directly to provide "extension" capabilities to an existing [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html). If you are the provider for a set of entities then you will want to implement [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html), this interface is primarily for extending an existing entity provider (adding extra functionality to one that is already registered Usage:  
1) Implement this interface  
2) Implement any additional capabilities interfaces (optional, but it would be crazy not to do at least one)  
3) Create a spring bean definition in the Sakai application context (components.xml)  
4) Implement AutoRegisterEntityProvider or register this implementation some other way

Methods:

### getEntityPrefix

java.lang.String getEntityPrefix()

Controls the globally unique prefix for the entities handled by this provider  
For example: Announcements might use "annc", Evaluation might use "eval" (if this is not actually unique then an exception will be thrown when Sakai attempts to register this broker)  
(the global reference string will consist of the entity prefix and the local id)

Returns:

the string that represents the globally unique prefix for an entity type

## Interface: EntityProviderManager

public interface EntityProviderManager

Handles all internal work of managing and working with the entity providers  
  
Registration of entity brokers happens via spring, see the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface for details

Methods:

### getEntityProperties

[EntityPropertiesService](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\providers\EntityPropertiesService.html) getEntityProperties()

Allows access to the current EntityPropertiesService service

Returns:

the current EntityPropertiesService service

### getEntityProviderMethodStore

[EntityProviderMethodStore](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProviderMethodStore.html) getEntityProviderMethodStore()

Allows access to the current EntityProviderMethodStore service

Returns:

the current EntityProviderMethodStore service

### getPrefixCapabilities

java.util.List<java.lang.Class<? extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)>> getPrefixCapabilities(java.lang.String prefix)

Get all the capabilities for a given entity prefix, WARNING: This is very inefficient so you should normally use [getProviderByPrefixAndCapability(String, Class)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProviderManager.html#getProviderByPrefixAndCapability%28java.lang.String,%20java.lang.Class%29) when trying to determine if a provider implements a specific capability

Parameters:

prefix - the string which represents a type of entity handled by an entity provider

Returns:

a list of the capabilities classes implemented by the entity provider defining this prefix

### getPrefixesByCapability

<T extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)> java.util.List<java.lang.String> getPrefixesByCapability(java.lang.Class<T> capability)

Gets the prefixes which support a specific capability

Type Parameters:

T - a class which extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Parameters:

capability - any entity provider capability class (these classes extend [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) or [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html) or another capability)

Returns:

the list of entity prefixes which support this capability, may be an empty list

### getProviderByPrefix

[EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) getProviderByPrefix(java.lang.String prefix)

Get the entity provider by the prefix which uniquely defines all entities of a type,   
NOTE: this returns the [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html) that handles the exists check (it may handle many other things as well), the basic [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) if there is no [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html), OR null if neither exists

Parameters:

prefix - the string which represents a type of entity handled by an entity provider

Returns:

the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) which handles this entity or null if none exists (only if prefix is not used)

### getProviderByPrefixAndCapability

<T extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)> T getProviderByPrefixAndCapability(java.lang.String prefix,

java.lang.Class<T> capability)

Get the entity provider by the prefix which uniquely defines all entities of a type and also handles a specific capability   
NOTE: this returns the provider that handles this capability (it may handle many other things as well)

Type Parameters:

T - a class which extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Parameters:

prefix - the string which represents a type of entity handled by an entity provider

capability - any entity provider capability class (these classes extend [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) or [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html) or another capability)

Returns:

the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) which handles this capability for this prefix or null if none exists or the prefix is not used

### getProvidersByCapability

<T extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)> java.util.List<T> getProvidersByCapability(java.lang.Class<T> capability)

Get all the entity providers which support a specific capability, this useful if you need to get the providers and call the capability methods on them directly

Type Parameters:

T - a class which extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Parameters:

capability - any entity provider capability class (these classes extend [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) or [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html) or another capability)

Returns:

the list of providers which implement the given capability, may be an empty list

### getRegisteredEntityCapabilities

java.util.Map<java.lang.String,java.util.List<java.lang.Class<? extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)>>> getRegisteredEntityCapabilities()

Get all registered prefixes and their capabilities, WARNING: This is very inefficient so you should normally use [getProviderByPrefixAndCapability(String, Class)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProviderManager.html#getProviderByPrefixAndCapability%28java.lang.String,%20java.lang.Class%29) when trying to determine if a provider implements a specific capability

Returns:

a map of prefix -> List(capabilities)

### getRegisteredPrefixes

java.util.Set<java.lang.String> getRegisteredPrefixes()

Retrieve a complete list of all currently registered [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)s

Returns:

all currently registered entity providers

### getRequestGetter

[RequestGetterWrite](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestGetterWrite.html) getRequestGetter()

Allows access to the current RequestGetter service

Returns:

the current RequestGetter service

### getRequestStorage

[RequestStorageWrite](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestStorageWrite.html) getRequestStorage()

Allows access to the current RequestStorage service

Returns:

the current RequestStorageWrite service

### registerEntityProvider

void registerEntityProvider([EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) entityProvider)

Registers an entity provider with the manager, this allows registration to happen programatically but the preferred method is to use the [AutoRegisterEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\AutoRegisterEntityProvider.html) instead (see the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface), replaces an existing entity provider which uses the same prefix and handles the same capabilities if one is already registered, does not affect other providers which handle the same prefix but handle other capabilities  
NOTE: This allows developers to register providers from all over the code base without requiring all capabilities to live in the same project (i.e. allows for a large reduction in dependencies and conflicts)

Parameters:

entityProvider - an entity provider to register with the main entity provider manager

### registerListener

<T extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)> void registerListener([EntityProviderListener](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityProviderListener.html)<T> listener,

boolean includeExisting)

Registers a listener which is called whenever entity providers are registered depending on the filters in the [EntityProviderListener](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityProviderListener.html)  
This is particularly useful for capabilities which should/must be executed one time only and should not be called over and over OR will not do anything until they are triggered

Parameters:

notifier - the entity provider listener to register

includeExisting - if true then the listener will be called for existing registered providers, otherwise it will only be called for newly registered providers

Throws:

java.lang.IllegalArgumentException - if the params are null

### unregisterCapability

void unregisterCapability(java.lang.String prefix,

java.lang.Class<? extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)> capability)

Unregisters an entity provider with the manager based on a prefix and capability, this will remove a registered entity broker from the manager registration, if the prefix and capability are not registered then no error is thrown  
NOTE: Attempting to unregister the base [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) will cause an exception, if you want to completely unregister a type of entity you must use the [unregisterEntityProviderByPrefix(String)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProviderManager.html#unregisterEntityProviderByPrefix%28java.lang.String%29)

Parameters:

prefix - the string which represents a type of entity handled by an entity provider

capability - any entity provider capability class (these classes extend [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) or [CoreEntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html) or another capability)

### unregisterEntityProvider

void unregisterEntityProvider([EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) entityProvider)

Unregisters an entity provider with the manager, this will remove a registered entity broker from the manager registration, if the entity provider supplied is not registered then no error is thrown

Parameters:

entityProvider - an entity provider to unregister with the main entity provider manager

### unregisterEntityProviderByPrefix

void unregisterEntityProviderByPrefix(java.lang.String prefix)

Unregisters an entity provider with the manager based on the entity prefix it handles, this will remove all registered entity providers from the manager registration by the prefix, if the entity provider prefix provided is not registered then no error is thrown  
This effectively purges the entire set of entity providers for a prefix

Parameters:

prefix - the string which represents a type of entity handled by entity providers

### unregisterListener

<T extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)> void unregisterListener([EntityProviderListener](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityProviderListener.html)<T> listener)

Unregisters the listener if it is registered or does nothing

Parameters:

notifier - the entity provider listener to unregister

Throws:

java.lang.IllegalArgumentException - if the params are null

## Interface: EntityProviderMethodStore

public interface EntityProviderMethodStore

This is the interface for handling storage of methods (related to custom actions and the like), this is for internal usage only and should not be accessed or used by other developers

Methods:

### addCustomAction

void addCustomAction(java.lang.String prefix,

[CustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\CustomAction.html) customAction)

Add a custom action for a prefix

Parameters:

prefix - an entity prefix

customAction - the custom action to add

### addURLRedirects

void addURLRedirects(java.lang.String prefix,

[URLRedirect](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\URLRedirect.html)[] redirects)

Add all URL redirects to the following prefix, maintains any existing ones

Parameters:

prefix - an entity prefix

redirects - an array of redirects

Throws:

java.lang.IllegalArgumentException - if any of the URL redirects are invalid

### findCustomActions

[CustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\CustomAction.html)[] findCustomActions([EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) entityProvider,

boolean ignoreFailures)

Get all the custom actions that can be found

Parameters:

entityProvider - the provider to search for custom actions

ignoreFailures - if true then will not throw exceptions if methods are not found

Returns:

the array of CustomAction objects which are found

### findURLRedirectMethods

[URLRedirect](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\URLRedirect.html)[] findURLRedirectMethods([EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) entityProvider)

Looks for redirect methods in the given entity provider

Parameters:

entityProvider - an entity provider

Returns:

an array of redirect objects

Throws:

java.lang.IllegalArgumentException - if the methods are setup incorrectly

### getCustomAction

[CustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\CustomAction.html) getCustomAction(java.lang.String prefix,

java.lang.String action)

Get the [CustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\CustomAction.html) for a prefix and action if it exists

Parameters:

prefix - an entity prefix

action - an action key

Returns:

the custom action OR null if none found

### getCustomActions

java.util.List<[CustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\CustomAction.html)> getCustomActions(java.lang.String prefix)

Gets the list of all custom actions for a prefix

Parameters:

prefix - an entity prefix

Returns:

a list of CustomActions for this prefix, empty if there are none

### getURLRedirects

java.util.List<[URLRedirect](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\URLRedirect.html)> getURLRedirects(java.lang.String prefix)

Get the list of all redirects for this prefix

Parameters:

prefix - the entity prefix

Returns:

a list of url redirects, may be empty if there are none

### removeCustomActions

void removeCustomActions(java.lang.String prefix)

Remove any custom actions that are set for this prefix

Parameters:

prefix - an entity prefix

### removeURLRedirects

void removeURLRedirects(java.lang.String prefix)

Remove any and all redirects for this prefix

Parameters:

prefix - an entity prefix

### setCustomActions

void setCustomActions(java.lang.String prefix,

java.util.Map<java.lang.String,[CustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\CustomAction.html)> actions)

Set the custom actions for this prefix

Parameters:

prefix - an entity prefix

actions - a map of action -> [CustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\CustomAction.html)

Package: org.sakaiproject.entitybroker.entityprovider.capabilities

Interface: ActionsDefineable

public interface ActionsDefineable

extends [ActionsExecutable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\ActionsExecutable.html)

This entity supports custom actions (as defined by RoR and REST microformat: http://microformats.org/wiki/rest/urls)  
This is the more controllable version, use [ActionsExecutable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\ActionsExecutable.html) if you want to use the conventions and allow the system to detect your custom actions based on method names and annotations  
This means that there are custom actions which can be invoked on entities or entity spaces, custom actions can augment the current entity operation or they can completely change the behavior and skip the current operation entirely  
You can create methods in your entity provider which either end with or use the [EntityCustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\annotations\EntityCustomAction.html) suffix to define the custom actions  
You can describe the actions using the [Describeable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Describeable.html) key: .action. = description  
If you want more control then you can use [ActionsExecutionControllable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\ActionsExecutionControllable.html)

Methods:

### defineActions

[CustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\CustomAction.html)[] defineActions()

Defines the custom actions which are allowed to be performed on your entities

Returns:

an array of the custom actions in the order they should be checked for in incoming requests  
The action keys should match with public methods in your provider which end with [ActionsExecutable.ACTION\_METHOD\_SUFFIX](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\ActionsExecutable.html#ACTION_METHOD_SUFFIX) or have the [EntityCustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\annotations\EntityCustomAction.html) annotation on them, use the fields in [CustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\CustomAction.html) to define the methodName

Interface: ActionsExecutable

public interface ActionsExecutable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

This entity supports custom actions (as defined by RoR and REST microformat: http://microformats.org/wiki/rest/urls)  
This means that there are custom actions which can be invoked on entities or entity spaces, custom actions can augment the current entity operation or they can completely change the behavior and skip the current operation entirely  
You can create methods in your entity provider which either end with ["CustomAction"](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\ActionsExecutable.html#ACTION_METHOD_SUFFIX) or use the [EntityCustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\annotations\EntityCustomAction.html) suffix to define the custom actions  
You can describe the actions using the [Describeable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Describeable.html) key: .action. = description  
If you want more control then you can use [ActionsDefineable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\ActionsDefineable.html) and [ActionsExecutionControllable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\ActionsExecutionControllable.html)

Methods:

### executeActions

java.lang.Object executeActions([EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) entityView,

java.lang.String action,

java.util.Map<java.lang.String,java.lang.Object> actionParams,

java.io.OutputStream outputStream)

This allows the developer to define how to execute custom actions on entities, this method will be called every time a custom action execution is requested, the incoming data provides the context for the action to be executed  
NOTE: The return data can be complex so please read carefully, entity data is returned as the default for the request if no format is specified

Parameters:

entityView - an entity view, should contain all the information related to the incoming entity request or URL, includes the entity reference and the requested format information

action - key which will be used to trigger the action (e.g. promote), will be triggered by a URL like so: /user/aaronz/promote

requestValues - this is an array which contains passed in action params, if this is running as a result of an http request this will include all the request variables, otherwise this will just return any custom values needed to execute this action

outputStream - an OutputStream to place binary or long text data into, if this is used for binary data then the [ActionReturn](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\ActionReturn.html) should be returned with the correct encoding information and the output variable set to the OutputStream

Returns:

this should return one of the following:   
1) null (this is ok in most circumstances to indicate the method is done, use an exception to indicate failure)   
2) an [ActionReturn](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\ActionReturn.html) (this is a special object used to indicate return states and handle binary data)   
3) a UTF-8 encoded OutputStream or String   
4) a List of entity objects   
5) an entity object   
6) a boolean value (true indicates success and is the same as returning null, false indicates failure and causes an [EntityNotFoundException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityNotFoundException.html)   
  
Note: Can throw the indicated exceptions and have them translated and handled, all others will pass through Can throw the following exceptions and have them translated and handled, all others will pass through:  
[EntityNotFoundException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityNotFoundException.html) to indicate the entity request could not find the data that was requested   
IllegalArgumentException to indicate that the incoming params or the request was invalid   
[FormatUnsupportedException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\FormatUnsupportedException.html) to indicate that the requested format is not supported for this entity request   
[EntityException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityException.html) to indicate a specific entity failure occurred, can include a response code and error message   
SecurityException to indicate that the the current user is no allowed to perform this action   
IllegalStateException to indicate a general failure has occurred

Interface: AutoRegisterEntityProvider

public interface AutoRegisterEntityProvider

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), [AutoRegister](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\collector\AutoRegister.html)

By implementing this interface you are telling the EntityProviderManager to register this entity broker as soon as spring creates it, to be exposed as part of the EntityBroker   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Interface: Browseable

public interface Browseable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

This indicates that this entity will participate in browse functionality for entities, For example, it will provide lists of entities which are visible to users in locations which can be looked through and selected  
This is the root interface for browsing entities and does not do anything by itself, you must add at least [BrowseableCollection](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\BrowseableCollection.html) or [BrowseSearchable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\BrowseSearchable.html) interface to this   
If your entities are nested then the nested ones will need to implement [BrowseNestable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\BrowseNestable.html)   
Entities which do not implement this will not appear in lists of entities which are being browsed   
Internationalization keys:  
{prefix}.browse = the name to show in the browse list for this entity   
{prefix}.browse.description = the optional description that is viable to show in the browse list for this entity   
  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Interface: BrowseableCollection

public interface BrowseableCollection

extends [Browseable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Browseable.html), [CollectionResolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\CollectionResolvable.html)

This indicates that this entity will participate in browse functionality for entities, For example, it will provide lists of entities which are visible to users in locations which can be looked through and selected  
Entities which do not implement this will not appear in lists of entities which are being browsed  
This is the convention interface and simply uses the results of calls to [CollectionResolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\CollectionResolvable.html) to provide lists of entities for browsing, the implementor should be sure that ordering and limiting are supported for the provided search object in their implementation of [CollectionResolvable.getEntities(org.sakaiproject.entitybroker.EntityReference, org.sakaiproject.entitybroker.entityprovider.search.Search)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\CollectionResolvable.html#getEntities%28org.sakaiproject.entitybroker.EntityReference,%20org.sakaiproject.entitybroker.entityprovider.search.Search%29)  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface  
See [Browseable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Browseable.html) for the i18n keys   
This extends [CollectionResolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\CollectionResolvable.html), use the [BrowseSearchable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\BrowseSearchable.html) interface if you require more control

Interface: BrowseNestable

public interface BrowseNestable

extends [Browseable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Browseable.html)

This indicates that this entity will participate in browse functionality for entities and it is nested within another entity: Example:   
A blog (parent) contains blog entries (children). When browsing a blog there is a need to be able to select the entries within the blog, the provider for the entry would implement this interface to allow the set of entries to be chosen within the blog   
The blog provider would not need to implement anything extra   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### getChildrenEntities

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> getChildrenEntities([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) parentRef,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search,

java.lang.String userReference,

java.lang.String associatedReference,

java.util.Map<java.lang.String,java.lang.Object> params)

Returns the list of entities being browsed which are nested under a parent entity with a given reference   
Example: Parent entity is /blog/123, there are 3 entries in this blog, the search limits the return to 2 entities This method should return the first 2 blog entries in the blog with reference /blog/123

Parameters:

parentRef - the reference object for the parent entity

search - (optional) a search object which can define the order to return entities, search filters, and total number of entities returned, may be empty but will not be null, implementors are encouraged to support ordering and limiting of the number of returned results at least

userReference - (optional) the unique entity reference for a user which is browsing the results, this may be null to indicate that only items which are visible to all users should be shown

associatedReference - (optional) a globally unique reference to an entity, this is the entity that the returned browseable data must be associated with (e.g. limited by reference to a location/site/group or other associated entity), this may be null to indicate there is no association limit

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

Returns:

a list of entity data objects which contain the reference, URL, display title and optionally other entity data

Throws:

java.lang.SecurityException - if the data cannot be accessed by the current user or is not publicly accessible

java.lang.IllegalArgumentException - if the reference is invalid or the search is invalid

java.lang.IllegalStateException - if any other error occurs

Interface: BrowseSearchable

public interface BrowseSearchable

extends [Browseable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Browseable.html)

This indicates that this entity will participate in browse functionality for entities, For example, it will provide lists of entities which are visible to users in locations which can be looked through and selected  
This provides fine grained control over which entities will appear in a browse list, normally [CollectionResolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\CollectionResolvable.html) should show all entities, however, for the browse list we will explicitly filter by users and/or locations and may not show all entities, entities which do not implement this or [BrowseableCollection](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\BrowseableCollection.html) will not appear in lists of entities which are being browsed  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface  
This is the configuration interface for [BrowseableCollection](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\BrowseableCollection.html) (the convention interface)

Methods:

### browseEntities

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> browseEntities([Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search,

java.lang.String userReference,

java.lang.String associatedReference,

java.util.Map<java.lang.String,java.lang.Object> params)

Returns the list of entities which are being browsed based on the given parameters   
See [Browseable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Browseable.html) for the i18n keys

Parameters:

search - a search object which can define the order to return entities, search filters, and total number of entities returned, may be empty but will not be null, implementors are encouraged to support ordering and limiting of the number of returned results at least

userReference - (optional) the unique entity reference for a user which is browsing the results, this may be null to indicate that only items which are visible to all users should be shown

associatedReference - (optional) a globally unique reference to an entity, this is the entity that the returned browseable data must be associated with (e.g. limited by reference to a location/site/group or other associated entity), this may be null to indicate there is no association limit

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

Returns:

a list of entity data objects which contain the reference, URL, display title and optionally other entity data

Throws:

java.lang.SecurityException - if the data cannot be accessed by the current user or is not publicly accessible

java.lang.IllegalArgumentException - if the reference is invalid or the search is invalid

java.lang.IllegalStateException - if any other error occurs

Interface: CollectionResolvable

public interface CollectionResolvable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), [Resolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Resolvable.html)

This type of entity supports retrieval of entities in a collection based on a search, this will be invoked when an entity space is accessed (/prefix) rather than accessing an individual entity (/prefix/id)  
The data is returned as a list of entity objects (Object, Map, whatever POJO, etc.) OR as a list of [EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html) objects (which can contain the entities or just information about them like properties, url, etc. This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### getEntities

java.util.List<?> getEntities([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search)

Allows these entities to be fetched based on search parameters, this should never return null and if there are no entities then the list should be empty  
Note: The entity class types in the list need to be able to be resolved from the ClassLoader of the EntityBrokerManager (currently this means deployed into shared)  
  
These do not have to be model objects and may simply be something created (e.g. String, Map, etc.) to give to anyone calling this method

Parameters:

ref - the parsed reference object which uniquely represents this entity, only the prefix will be used from this reference (since that identifies the space and collection)

search - a search object which can define the order to return entities, search filters, and total number of entities returned,  
NOTE: There are some predefined search keys which you may optionally support, provider are encourage to support the SEARCH\_\* search keys listed in this interface

Returns:

a list of entity objects (POJOs, Map, etc.) of the type handled by this provider OR a list of [EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html) objects based on the search OR empty if none found, should not return null

Throws:

java.lang.SecurityException - if the data cannot be accessed by the current user or is not publicly accessible

java.lang.IllegalArgumentException - if the reference is invalid or the search is invalid

java.lang.IllegalStateException - if any other error occurs

Interface: Createable

public interface Createable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), [Sampleable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Sampleable.html)

This entity type can be created (this is the C in CRUD), the current user id should be used for permissions checking in most cases  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### createEntity

java.lang.String createEntity([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref,

java.lang.Object entity,

java.util.Map<java.lang.String,java.lang.Object> params)

Create a new entity and return the unique local id of the entity, the object should contain the data needed to create the entity or this will fail

Parameters:

ref - the parsed reference object which uniquely represents this entity

entity - an entity object

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

Returns:

the locally unique id of the new object

Throws:

java.lang.IllegalArgumentException - if the entity could not be created because of missing or invalid data

java.lang.SecurityException - if permissions prevented this entity from being created

java.lang.IllegalStateException - for all other failures

Interface: CRUDable

public interface CRUDable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), [Saveable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Saveable.html), [Createable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Createable.html), [Resolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Resolvable.html), [Updateable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Updateable.html), [Deleteable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Deleteable.html)

A convenient interface to define that this entity type supports all CRUD operations  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Interface: Deleteable

public interface Deleteable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

This entity type can be deleted (this is the D in CRUD), the current user id should be used for permissions checking in most cases  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### deleteEntity

void deleteEntity([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref,

java.util.Map<java.lang.String,java.lang.Object> params)

Deletes the entity identified by this entity reference, if the entity cannot be found then nothing happens

Parameters:

ref - the parsed reference object which uniquely represents this entity

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

Throws:

java.lang.SecurityException - if permissions prevented this entity from being created

java.lang.IllegalStateException - for all other failures

Interface: Describeable

public interface Describeable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

This entity will describe itself, this description can be accessed for display in interfaces, the description should generally be targeted at developers (in particular, those using REST or web services)  
This is the convention interface, it will simply look for the properties file in the classloader which your [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) is located in, the file must be named <entity-prefix>.properties (e.g. myentity.properties)  
The keys inside the file must be as follows:  
<entity-prefix> = This is the main description of the entity, appears at the top   
<entity-prefix>.view.<viewKey> = This is a description about a particular view for an entity (viewKey from entity view constants [EntityView.VIEW\_LIST](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html#VIEW_LIST))   
<entity-prefix>.action.<actionKey> = This is a description of a custom action for this entity (see [ActionsExecutable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\ActionsExecutable.html) or [EntityCustomAction](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\annotations\EntityCustomAction.html))   
<entity-prefix>.field.<fieldName> = This is a description about a particular entity field for this entity object (see [Resolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Resolvable.html) or [CollectionResolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\CollectionResolvable.html))   
<entity-prefix>.redirect.<redirectTemplate> = This is a description about a particular redirect rule for this entity URL space (see [Redirectable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Redirectable.html) or [EntityURLRedirect](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\annotations\EntityURLRedirect.html))   
<entity-prefix>.<capability> = This is a description about a particular capability for this entity   
Example:

myentity = This is my entity, it is used for <b>examples</b> only

myentity.view.show = this shows a single instance of my entity

myentity.action.copy = this makes a copy of an instance of a myentity

myentity.field.name = this is the name of the entity, it is a user displayable name

myentity.redirect./{prefix}/xml/{id} = redirects to an xml view of a myentity

myentity.Inputable = <i>extra</i> notes about the Inputable implementation for myentity

If you need to define the location of your properties file then use: [DescribePropertiesable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\DescribePropertiesable.html)   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Interface: DescribeDefineable

public interface DescribeDefineable

extends [Describeable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Describeable.html)

Allows an entity to define the description of itself in code rather than using properties, this will be called each time a description is needed so it should be efficient  
This is the configuration interface  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### getDescription

java.lang.String getDescription(java.util.Locale locale,

java.lang.String descriptionKey)

Allows for complete control over the descriptions of entities  
This will always be called first if it is defined, returning a null will default to attempting to get the value from the properties (if any are defined), returning an empty string will cause nothing to be shown for the description

Parameters:

locale - this is the locale that the description should be created for

descriptionKey - (optional) if null then the general description of the entity should be created, otherwise provide the description for the capability that was provided (e.g. Resolveable) OR the custom action, starts with action.<actionKey> (e.g. action.promote), see the [Describeable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Describeable.html) interface for information about the other keys that will be passed in

Returns:

the string which describes this entity or this capability for this entity OR '' for no description OR return null to allow this to attempt to get the value from the properties file

Interface: DescribePropertiesable

public interface DescribePropertiesable

extends [Describeable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Describeable.html)

This allows an entity to define the Properties files/classes to be used for looking up descriptions of the entity ([Describeable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Describeable.html))  
This is the configuration interface  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### getBaseName

java.lang.String getBaseName()

Defines the Properties file/class baseName which is used to find descriptions of an entity as per the keys defined in [Describeable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Describeable.html)  
The baseName will be the file without the .properties or the locale codes  
Examples:  
myentity - would match: myentity.properties, myentity\_en.properties, myentity\_en\_GB.properties, etc.  
mydir/myloc/mything - would match: mydir/myloc/mything.properties, etc.

Returns:

the baseName of the properties file/classes in the [getResourceClassLoader()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\DescribePropertiesable.html#getResourceClassLoader%28%29)

### getResourceClassLoader

java.lang.ClassLoader getResourceClassLoader()

Defines the ClassLoader which will be used to load the properties bundle file/classes which contain the entity descriptions

Interface: EntityViewUrlCustomizable

public interface EntityViewUrlCustomizable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Indicates that the entity URLs for the types of entities handled by this provider are customized by the set of entity templates returned  
Methods:

### getParseTemplates

java.util.List<[TemplateParseUtil.Template](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.Template.html)> getParseTemplates()

Defines a set of parseable templates to use with entity url processing for this template (which is basically a key and the template string), the array which defines the set of template keys is [TemplateParseUtil.PARSE\_TEMPLATE\_KEYS](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#PARSE_TEMPLATE_KEYS)  
Rules for parse templates:  
1) "{","}", and #SEPARATOR are special characters and must be used as indicated only  
2) Must begin with a #SEPARATOR, must not end with a #SEPARATOR  
3) must begin with "/{prefix}" (use the #SEPARATOR and #PREFIX constants)  
3) each {var} can only be used once in a template  
4) {var} can never touch each other (i.e /{var1}{var2}/{id} is invalid)  
5) each {var} can only have the chars from [TemplateParseUtil.VALID\_VAR\_CHARS](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#VALID_VAR_CHARS)  
6) parse templates can only have the chars from [TemplateParseUtil.VALID\_TEMPLATE\_CHARS](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#VALID_TEMPLATE_CHARS)  
7) Empty braces ({}) cannot appear in the template  
  
You do not have to supply a template for all the keys in [TemplateParseUtil.PARSE\_TEMPLATE\_KEYS](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#PARSE_TEMPLATE_KEYS), any that you do not include will simply use the default templates, be careful though, since they will be parsed in order you have to be careful about the order you place your templates in the list, check the default order as an example

Returns:

the list of custom parsing templates in the order they should be processed

Interface: Exportable

public interface Exportable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Indicates an entity provider has the capability of exporting entity data which is related to other entities, note that the decision about which data to export is left up to the implementor based on the reference supplied   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### exportData

java.lang.String exportData(java.lang.String reference,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search,

java.io.OutputStream data,

boolean destructive,

java.util.Map<java.lang.String,java.lang.Object> params)

Request an export stream of data from an entity provider for all data related to a specific entity (this will probably not be an entity in this provider), search, and parameters  
This is primarily to support the use case archiving and exporting data from a system which related to a user or a site/group

Parameters:

reference - a globally unique reference to an entity, this is the entity that the exported data should be associated with (e.g. a reference to a site object or user)

search - (optional) a search which should be used to limit the data which is exported, may be null

data - a stream to put the export data into which will be saved by the archiver/exporter

destructive - if false then the data being exported is not changed, if true then the data should be deleted or hidden (depending on the internal operation of the entity)

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

Returns:

a string key representing the encoding used and possibly other info like a version, this allows the export to provide tips to the import when data is streamed back in, if there is no data to export then a null will be returned

Interface: Importable

public interface Importable

extends [Exportable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Exportable.html)

Indicates an entity provider has the capability of importing entity data that was previously exported via the [Exportable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Exportable.html) capability which will be related to other entities, note that the way to associate the data is left up to the implementor based on the reference supplied   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### importData

java.lang.String[] importData(java.lang.String reference,

java.io.InputStream data,

java.lang.String encodingKey,

java.util.Map<java.lang.String,java.lang.Object> params)

Request that an import stream be turned into real data related to the entities in this provider and associated with a specific entity (this will probably not be an entity in this provider)

Parameters:

reference - a globally unique reference to an entity, this is the entity that the imported data should be associated with (e.g. a reference to a site object or user)

data - a stream of data from the archiver/importer, this should match a previous export stream exactly (or at least very closely in order for the importer to be able to understand it)

encodingKey - a string representing the encoding used and possibly other info like a version, this should be the string sent with the export

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

Returns:

the array of all entity references which were created from the import OR empty if none created

Throws:

java.lang.IllegalArgumentException - if any arguments are invalid or missing

java.lang.IllegalStateException - if a failure occurs with the import (message should be as descriptive as possible)

Interface: Inputable

public interface Inputable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html), [Sampleable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Sampleable.html)

These entities can be entered as certain input formats which are handled automatically and translated into entity objects (of the type defined by [Sampleable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Sampleable.html))  
If you want to define the way the input is translated instead of using the internal methods then use [InputTranslatable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\InputTranslatable.html)

Methods:

### getHandledInputFormats

java.lang.String[] getHandledInputFormats()

Defines the input format types (extensions) handled by this provider  
NOTE: In the case of an entity view the extension which goes on the end of an entity URL (after a ".") indicates the input type  
WARNING: not including [Formats.HTML](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html#HTML) in the return will stop all redirects to the access providers and therefore will cause HTML requests for entities to go nowhere

Returns:

an array containing the extension formats (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html)) handled, use the constants (example: [Formats.XML](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html#XML)) or feel free to make up your own if you like

Interface: InputTranslatable

public interface InputTranslatable

extends [Inputable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Inputable.html)

Allows this entity to define the way to translate data for a reference into an entity object depending on the format requested, if you just want to use the internal methods to handle formatting the input into an entity then simply use [Inputable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Inputable.html)  
NOTE: throwing [FormatUnsupportedException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\FormatUnsupportedException.html) will pass control over to the internal handlers for formatting, if you want to stop the request for this format type entirely then throw an IllegalStateException and the processing will be halted

Methods:

### translateFormattedData

java.lang.Object translateFormattedData([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref,

java.lang.String format,

java.io.InputStream input,

java.util.Map<java.lang.String,java.lang.Object> params)

Translates the input data stream in the supplied format into an entity object for this reference

Parameters:

ref - the parsed reference object which uniquely represents this entity

format - a string constant indicating the extension format (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html)) of the input, (example: [Formats.XML](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html#XML))

input - an stream which contains the data to make up this entity, you may assume this is UTF-8 encoded if you don't know anything else about it

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

Returns:

an entity object of the type used for these entities

Throws:

[FormatUnsupportedException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\FormatUnsupportedException.html) - if you do not handle this format type (passes control to the internal handlers)

[EntityEncodingException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityEncodingException.html) - if you cannot encode the received data into an entity

java.lang.IllegalArgumentException - if any of the arguments are invalid

java.lang.IllegalStateException - for all other failures

Interface: Outputable

public interface Outputable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html)

These entities can be returned as certain output formats which are handled automatically  
If you want to define the data that is returned instead of using the internal methods then use [OutputFormattable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\OutputFormattable.html)  
  
NOTE: By default all entity view requests go through to the available access providers: [EntityViewAccessProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProvider.html) or HttpServletAccessProvider NOTE: there is no internal handling of HTML, it will always redirect to the the available access provider if there is one (if there is not one then the entity will be toStringed)

Methods:

### getHandledOutputFormats

java.lang.String[] getHandledOutputFormats()

Defines the output format types (extensions) handled by this provider  
NOTE: In the case of an entity view the extension which goes on the end of an entity URL (after a ".") indicates the return type  
WARNING: This combines with the access interface when http requests are being processed, all requests will pass through to the [EntityViewAccessProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProvider.html) if they are not handled

Returns:

an array containing the extension formats (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html)) handled   
OR empty array to indicate all are handled (note that the internal formatter will throw exceptions when it cannot handle a type)   
OR null to indicate none are handled (same as not implementing AccessFormats)   
NOTE: use the constants (example: [Formats.HTML](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html#HTML)) or feel free to make up your own if you like

Interface: OutputFormattable

public interface OutputFormattable

extends [Outputable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Outputable.html)

Allows this entity to define the output data format for a reference or a list of entity objects depending on the format requested, if you just want to use the internal methods to handle formatting the output then simply use [Outputable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Outputable.html)  
NOTE: throwing [FormatUnsupportedException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\FormatUnsupportedException.html) will pass control over to the internal handlers for formatting, if you want to stop the request for this format type entirely then throw an IllegalStateException and the processing will be halted

Methods:

### formatOutput

void formatOutput([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref,

java.lang.String format,

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> entities,

java.util.Map<java.lang.String,java.lang.Object> params,

java.io.OutputStream output)

Formats the entity or collection included or referred to by this entity ref object into output according to the format string provided, Should take into account the reference when determining what the entities are and how to encode them   
NOTE: be careful to correctly handle the list of entities which are meant to be encoded, note that the [EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html) objects include meta data and the data they contain can be of any object type (though all data will come from your provider so the types should not be surprising)

Parameters:

ref - the parsed reference object which uniquely represents this entity

format - a string constant indicating the extension format (from [Formats](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html)) for output, (example: [Formats.XML](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\Formats.html#XML))

entities - (optional) a list of entity data objects to create formatted output for, if this is null then the entities should be retrieved based on the reference, if this contains only a single item AND the ref refers to a single entity then the entity data object should be extracted from the list and encoded without the indication that it is a collection, for all other cases the encoding should include an indication that this is a list of entities

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

output - the output stream to place the formatted data in, should be UTF-8 encoded if there is char data

Throws:

[FormatUnsupportedException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\FormatUnsupportedException.html) - if you do not handle this format type (passes control to the internal handlers)

[EntityEncodingException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityEncodingException.html) - if you cannot format the entity data for some reason

java.lang.IllegalArgumentException - if any of the arguments are invalid

java.lang.IllegalStateException - for all other failures

Interface: OutputSerializable

public interface OutputSerializable

extends [Outputable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Outputable.html)

Allows this entity to better control the data that is going to be output by allowing it to intercept the entities and return them as whatever objects which should be serialized for output, if you just want to use the internal methods to handle formatting the output then simply use [Outputable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Outputable.html)

Methods:

### makeSerializableObject

java.lang.Object makeSerializableObject([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref,

java.lang.Object entity)

Take the entity and convert it to whatever objects (Map, List, String, etc.) that you want to output, this will be called every time that an entity is about to serialized but only if the type of object matches that of the entities handled by your provider

Parameters:

the - entity reference for the current entity

entity - an object of the type handled by your provider

Returns:

the object you want to be serialized

Interface: Propertyable

public interface Propertyable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Allows entities handled by the entity provider which implements this interface to have meta properties attached to them, properties can be accessed via the EntityBroker, properties will be stored and retrieved using the internal entity property retrieval implementation  
For our usage, meta properties are extra or additional properties that are attached to an entity at runtime but are persisted so they can be retrieved later  
WARNING: this should be used only for properties of entities which will be accessed very lightly, for production level access OR the ability to control how properties are stored on your own entities, you should use the [PropertyProvideable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\PropertyProvideable.html) instead   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Interface: PropertyProvideable

public interface PropertyProvideable

extends [Propertyable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Propertyable.html), [PropertiesProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\PropertiesProvider.html)

Allows entities handled by the entity provider which implements this interface to have meta properties attached to them, these properties can be accessed via the [EntityBroker](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityBroker.html), properties will be stored and retrieved via the methods which are implemented in this interface  
For our usage, meta properties are extra or additional properties that are attached to an entity at runtime but are persisted so they can be retrieved later  
Allows the entity provider to define and control the way properties are stored on its own entities, if you would prefer to use the internal storage and only have lightweight property storage needs then use [Propertyable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Propertyable.html) instead  
NOTE: the validity of references and parameters is checked in the broker before the call goes to the provider   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Interface: Redirectable

public interface Redirectable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

This entity type has the ability to define and handle configurable URLs, URLs like this can be handled and supported:  
/gradebook/7890/student/70987 to view all the grades for a student from a course   
/gradebook/6758/item/Quiz1 to view a particular item in a gradebook by it's human readable name   
/gradebook/item/6857657 to maybe just a view an item by its unique id.   
Works with the [EntityURLRedirect](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\annotations\EntityURLRedirect.html) annotation  
This is the convention interface  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Interface: RedirectControllable

public interface RedirectControllable

extends [Redirectable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Redirectable.html)

This entity type has the ability to define and handle configurable URLs  
This adds the ability to control all redirects via a central method  
URLs like this can be handled and supported:  
/gradebook/7890/student/70987 to view all the grades for a student from a course   
/gradebook/6758/item/Quiz1 to view a particular item in a gradebook by it's human readable name   
/gradebook/item/6857657 to maybe just a view an item by its unique id.   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface  
The convention interface is at [Redirectable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Redirectable.html), there is also a capability for handling simple redirects at [RedirectDefinable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\RedirectDefinable.html)

Methods:

### defineHandledTemplatePatterns

java.lang.String[] defineHandledTemplatePatterns()

Defines all the URL patterns that will be matched and passed through to #handleRedirects(String, String[], Map) NOTE: /{prefix}/ must be included as the start of the template

Returns:

a list of all the handled URL patterns

### handleRedirects

java.lang.String handleRedirects(java.lang.String matchedTemplate,

java.lang.String incomingURL,

java.lang.String[] incomingSegments,

java.util.Map<java.lang.String,java.lang.String> incomingVariables)

Explicitly handles all the incoming URLs which match the patterns given by [defineHandledTemplatePatterns()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\RedirectControllable.html#defineHandledTemplatePatterns%28%29) do some processing to turn it into an outgoing URL OR just do some processing OR indicate that a failure has occurred

Parameters:

matchedPattern - the template pattern that was matched

incomingURL - the incoming URL that was matched by a URL pattern

incomingSegments - incoming URL segments, Example: /prefix/123/apple => {'prefix','123','apple'}

incomingVariables - a map of the values in the {} (prefix is always included), Example: pattern: /prefix/{thing}/apple, url: /prefix/123/apple, would yield: 'thing' => '123'

Returns:

should be one of the following:   
1) the URL to redirect to, will be processed as an external redirect if it starts with "http" or "/" (unless it starts with "/{prefix}"), otherwise it will be processed as an internal forward   
2) "" (empty string) to not redirect and return an empty success response   
3) null to not redirect and allow standard processing of the URL to continue

Throws:

java.lang.IllegalStateException - if there is a failure, this will cause the server to return a redirect failure

Interface: RedirectDefinable

public interface RedirectDefinable

extends [Redirectable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Redirectable.html)

This entity type has the ability to define and handle configurable URLs  
This adds the ability to supply a large set of simple redirects  
URLs like this can be handled and supported:  
/gradebook/7890/student/70987 to view all the grades for a student from a course   
/gradebook/6758/item/Quiz1 to view a particular item in a gradebook by it's human readable name   
/gradebook/item/6857657 to maybe just a view an item by its unique id.   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface  
The convention interface is at [Redirectable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Redirectable.html)

Methods:

### defineURLMappings

[TemplateMap](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\TemplateMap.html)[] defineURLMappings()

Defines the set of simple URL rewrites for this prefix  
Simple rewrites require no processing logic to handle the redirect and the redirect is always processed before anything validity checks happen  
Some examples:  
/myprefix/item/{id} => /my-item/{id}   
/myprefix/{year}/{month}/{day} => /myprefix/?date={year}-{month}-{day}  
incomingURL is the URL template pattern to match including the /prefix using {name} to indicate variables   
Example: /{prefix}/{thing}/site/{siteId} will match the following URL:   
/myprefix/123/site/456, the variables will be {prefix => myprefix, thing => 123, siteId => 456}   
NOTE: all incoming URL templates must start with "/{prefix}" ([TemplateParseUtil.TEMPLATE\_PREFIX](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\util\TemplateParseUtil.html#TEMPLATE_PREFIX))   
outgoingURL is the URL template pattern to fill with values from the incoming pattern, this can start with anything, but will be processed as an external redirect if it starts with "http" or "/" (unless it starts with "/{prefix}"), otherwise it will be processed as an internal forward   
NOTE: the special variables which are available to all outgoing URLs from the system are:  
{prefix} = the entity prefix   
{extension} = the extension if one is available or '' if none   
{dot-extension} = the extension with a '.' prepended if one is set or '' if no extension   
{query-string} = the query string (e.g auto=true) or '' if none   
{question-query-string} = the query string with a '?' prepended (e.g ?auto=true) or '' if none

Returns:

the array of template mappings (incomingURL pattern => outgoingURL pattern) OR null/empty if you have no simple mappings

Interface: ReferenceParseable

public interface ReferenceParseable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Indicates an entity provider has the capability of parsing its own reference string. An entity that does not implement this interface is assumed to deal in references of type [EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) (/prefix/id or /prefix)   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### getParsedExemplar

[EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) getParsedExemplar()

Returns an example instance of the [EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) class that this [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) uses as its reference type. If you do not also implement ParseSpecParseable a default parse specification will be inferred for you (the entity prefix will always come first).  
NOTE: you will probably want to override at LEAST the main constructor and a few other methods, check the commenting in the [EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) for tips

Returns:

an entity reference class which must extend [EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html)

Interface: RequestAware

public interface RequestAware

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Indicates that this entity provider needs to be request aware, this allows the entity provider to get hold of information from the request at any time by directly accessing the request and response objects (if we are inside a request), if there is no current request then this method will fail to return anything  
This is primarily intended to provide access to request parameters while operating inside the entity provider

Methods:

### setRequestGetter

void setRequestGetter([RequestGetter](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestGetter.html) requestGetter)

Allows the entity provider to access the current request if it is available, sets a getter service which will retrieve the current request/response if there is one  
NOTE: this will only be the current request at the instant that the methods on the getter service are called

Interface: RequestHandler

public interface RequestHandler

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), [EntityViewAccessProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\access\EntityViewAccessProvider.html)

Indicates that this entity provider will handle its own entity view requests, this would be very unusual but it allows the entity provider itself to redirect requests to a tool and normally would be used if there is some special circumstance only  
WARNING: This will be called before any other request handling and before the access provider is called and will cause all other processing to be skipped (includes REST calls, custom actions, formatting, etc.)  
  
NOTE: if you want to stop certain requests from coming through then a better option is to use [RequestInterceptor](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\RequestInterceptor.html) which is triggered just before this would be called

Interface: RequestInterceptor

public interface RequestInterceptor

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Allows actions to be taken before a direct request is handled or after it has been handled, will only affect requests coming in via the direct servlet

Methods:

### after

void after([EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) view,

javax.servlet.http.HttpServletRequest req,

javax.servlet.http.HttpServletResponse res)

Take actions after the request is handled for an entity view, this will be called just before each response is sent back to the requester, normally this would be used to add something to the response as it is getting ready to be sent back to the requester

Parameters:

view - an entity view, should contain all the information related to the incoming entity URL

req - the servlet request (available in case you need to get anything out of it)

res - the servlet response, put the correct data response into the outputstream

### before

void before([EntityView](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityView.html) view,

javax.servlet.http.HttpServletRequest req,

javax.servlet.http.HttpServletResponse res)

Take actions before the request is handled for an entity view, this will be called just before each request is sent to the correct request handler, this might be used to add information to the response before it goes on to be handled or to take some action as a result of information in the request or reference,  
if you want to interrupt the handling of this request (stop it) then throw an [EntityException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityException.html) and include the type of response you would like to return in the exception (this can be a success or failure response status)

Parameters:

view - an entity view, should contain all the information related to the incoming entity URL

req - the servlet request (available in case you need to get anything out of it)

res - the servlet response, put the correct data response into the outputstream

Interface: RequestStorable

public interface RequestStorable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Indicates that this entity provider is aware of the requests and can get to the stored values in the request or can store its own, this allows the entity provider to get hold of information from the request at any time  
This is primarily intended to provide access to request data while operating inside the entity provider without depending on servlet knowledge  
This provides access to the special indicator values which can be used to see what kind of request is operating and get information about it, see the [RequestStorage](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestStorage.html) object for more info  
If you need to get to the servlet data see [RequestAware](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\RequestAware.html) and [RequestInterceptor](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\RequestInterceptor.html)

Methods:

### setRequestStorage

void setRequestStorage([RequestStorage](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestStorage.html) requestStorage)

Allows the entity provider to access the current request storage if available, sets a storage service which will retrieve or set the stored data values  
NOTE: this will only access data from the current request at the time the call is made, values disappear as soon as the request ends

Interface: Resolvable

public interface Resolvable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Allows the entities handled by this provider to be accessed directly as objects, this is also the interface for "reading" entities (this is the R in CRUD)  
This is also used for resolving the type of entities  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### getEntity

java.lang.Object getEntity([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref)

Allows this entity to be fetched based on the ref (prefix and local id), also used to determine the class type of these entities by requesting an entity without specifying an id (id = null)  
WARNING: this method should not return null, throw the appropriate exception if the entity data cannot be found Note: The entity class type needs to be able to be resolved from the ClassLoader of the EntityBrokerManager (currently this means deployed into shared)  
  
The entity object does not have to be a model object itself and may simply be something created (Map, String, [EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html), etc.) to give to anyone calling this method.

Parameters:

ref - the parsed reference object which uniquely represents an entity OR only a prefix (indicating you should return a default constructed object with no properties set)

Returns:

an entity object for the prefix and id in the provided ref OR an empty entity object if only the prefix is supplied (should NOT return null), this return could be an [EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html) object, Map, your own POJO, etc.

Throws:

java.lang.IllegalArgumentException - if the id is invalid for this type of entity, NOTE: a null id means you should return a default constructed object)

java.lang.SecurityException - if access to this entity is not allowed

java.lang.IllegalStateException - for all other errors

Interface: RESTful

public interface RESTful

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), [CRUDable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\CRUDable.html), [CollectionResolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\CollectionResolvable.html), [Outputable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Outputable.html), [Inputable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Inputable.html), [Describeable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Describeable.html), [ActionsExecutable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\ActionsExecutable.html), [Redirectable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Redirectable.html)

Indicates that entities handled by this provider are RESTful as defined by the REST microformat:  
<http://microformats.org/wiki/rest/urls>  
Requires all CRUD functionality, Collection handling, HTML handling, and ability to support output formats  
This is mostly a convenience interface to make sure that everything needed to support REST has been implemented, it also includes the marker to indicate that RESTful URLs should work for entities of this type

Interface: Sampleable

public interface Sampleable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

This entity type can be known and this allows access to a sample object which represents it, the sample object can be of any type including a POJO, a Map, etc.  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### getSampleEntity

java.lang.Object getSampleEntity()

Provides a sample entity object which can be populated with data and then passed to methods like [Createable.createEntity(EntityReference, Object, java.util.Map)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Createable.html#createEntity%28org.sakaiproject.entitybroker.EntityReference,%20java.lang.Object,%20java.util.Map%29), this is necessary so that the type of the entity object is known and the right fields can be pre-filled, it also allows us to support the case of different read and write objects Note: The entity class type needs to be able to be resolved from the ClassLoader of the [EntityBroker](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityBroker.html) (currently this means deployed into shared)

Returns:

a sample entity object for entities of the type represented by this provider

Throws:

java.lang.IllegalStateException - if the sample object cannot be obtained for some reason

Interface: Saveable

public interface Saveable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), [Createable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Createable.html), [Updateable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Updateable.html)

Convenience interface to indicates that an entity is can be saved, i.e. it is creatable and updateable

Interface: Searchable

public interface Searchable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Indicates an entity provider has the capability of allowing the entities associated with it to be searchable, entities which implement this will be linked to the core search functionality   
If you need to force an entity or entities to be added to the search indexes then you should use the [SearchProvider.add(String, org.sakaiproject.entitybroker.entityprovider.extension.SearchContent)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\SearchProvider.html#add%28java.lang.String,%20org.sakaiproject.entitybroker.entityprovider.extension.SearchContent%29) method in the system core to cause immediate indexing   
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### getAllRefs

java.util.List<java.lang.String> getAllRefs(java.lang.String context)

Retrieve all entities for a given context

Parameters:

context - probably a site or group but this may also be null to indicate that the entire list of all entity refs should be sent

Returns:

the list of all entity references (e.g. /myprefix/myid)

### getData

[SearchContent](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\SearchContent.html) getData(java.lang.String reference)

Will be called to retrieve the data for this item when the system thinks the data related to an entity may have changed

Parameters:

reference - a globally unique reference to an entity (e.g. /myprefix/myid), consists of the entity prefix and optional segments (normally the id at least)

Returns:

the search content which should be indexed OR null to cause nothing to happen

Throws:

[EntityNotFoundException](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\exception\EntityNotFoundException.html) - if it does not exist (causes the search system to purge out the entry)

Interface: Statisticable

public interface Statisticable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

This capability is for tracking statistics of events for entities related to a tool, it will be used by the site stats service for event tracking and reporting  
Contact Nuno Fernandes (nuno@ufp.edu.pt) if you have questions  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### getAssociatedToolId

java.lang.String getAssociatedToolId()

Return the associated common tool.id for this tool

Returns:

the tool id (example: "sakai.messages")

### getEventKeys

java.lang.String[] getEventKeys()

Return an array of all the event keys which should be tracked for statistics

Returns:

an array if event keys (example: "message.new" , "message.delete")

### getEventNames

java.util.Map<java.lang.String,java.lang.String> getEventNames(java.util.Locale locale)

OPTIONAL: return null if you do not want to implement this  
Return the event key => event name map for a given Locale, allows the author to create human readable i18n names for their event keys

Parameters:

locale - the locale to return the names for

Returns:

the map of event key => event name (example: for a 'en' locale: {"message.new","A new message"}) OR null to use the event keys

Interface: Taggable

public interface Taggable

extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)

Allows an entity to have tags associated with it which can be searched for or simply used as a way to link to this entity   
This uses the internal tag storage mechanism or a central tag storage mechanism to store the tag applications  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Interface: TagProvideable

public interface TagProvideable

extends [Taggable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Taggable.html), [TagProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\TagProvider.html)

Allows an entity to control and provide tag storage and searching, this overrides internal tag storage and any existing tag storage will be ignored in favor of the provided tags  
This is the provider interface for [Taggable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Taggable.html), inherits all methods from [TagProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\TagProvider.html)  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### findEntitesByTags

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> findEntitesByTags(java.lang.String[] tags,

boolean matchAll,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search)

Search for all entities which have the given tags, can limit the return using the search object  
NOTE: Don't forget to check the [RequestStorable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\RequestStorable.html) request params for extra information about the current user and location and other possible params when implementing this

Parameters:

tags - a set of tags associated with entities

matchAll - if true then all tags must exist on the entity for it to be matched, if false then the entity just has to have one or more of the given tags

search - (optional) a search object, used to order or limit the number of returned results, restrictions will be typically ignored

Returns:

a list of entity search results (contains the ref, url, displayname of the matching entities)

Throws:

java.lang.IllegalArgumentException - if the tags set is empty or null

Interface: Updateable

public interface Updateable

extends [Resolvable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Resolvable.html)

This entity type can be updated (this is the U in CRUD), the current user id should be used for permissions checking in most cases  
This is one of the capability extensions for the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) interface

Methods:

### updateEntity

void updateEntity([EntityReference](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\EntityReference.html) ref,

java.lang.Object entity,

java.util.Map<java.lang.String,java.lang.Object> params)

Update an existing entity, the object should contain the data needed to update the entity or this will fail  
Typically the entity will be retrieved first using [Resolvable.getEntity(EntityReference)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Resolvable.html#getEntity%28org.sakaiproject.entitybroker.EntityReference%29) and the the fields will be updated and it will be passed into this method

Parameters:

ref - the parsed reference object which uniquely represents this entity

entity - an entity object

params - (optional) incoming set of parameters which may be used to send data specific to this request, may be null

Throws:

java.lang.IllegalArgumentException - if the entity could not be updated because of missing or invalid data or could not find entity to update

java.lang.SecurityException - if permissions prevented this entity from being updated

java.lang.IllegalStateException - for all other failures

Package: org.sakaiproject.entitybroker.entityprovider.extension

## Interface: EntityProviderListener<T extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)>

public interface EntityProviderListener<T extends [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)>

This allows a developer to be notified whenever a provider is registered, it will also ensure that it calls the listener method for any providers that are already registered when it is first registered if desired  
Usage:  
Just implement this class and use EntityProviderManager#registerNotifier(EntityProviderNotifier) to register it with the system. Whenever an entity provider is registered the [run(EntityProvider)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityProviderListener.html#run%28T%29) method is called. You can limit the calls to your listener using the [getPrefixFilter()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityProviderListener.html#getPrefixFilter%28%29) and [getCapabilityFilter()](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityProviderListener.html#getCapabilityFilter%28%29).  
Note that you can return a null prefix filter and capability filter but it means you will get called for every registration of every provider. You will want to use the [EntityProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html) for the generics to receive every capability type.

Methods:

### getCapabilityFilter

java.lang.Class<[T](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityProviderListener.html)> getCapabilityFilter()

Returns:

the capability to limit notifications to OR null for any capability (still will only be called once per provider)

### getPrefixFilter

java.lang.String getPrefixFilter()

Returns:

the prefix to limit the notifications to OR null for any prefix

### run

void run([T](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityProviderListener.html) provider)

This method will be called once per each provider that is registered with the system depending on the filter methods in this class  
WARNING: Do not hold onto (cache) the provider returned to you by this method, if you need to use it again later you should fetch it using the methods in [EntityProviderManager](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProviderManager.html) like [EntityProviderManager.getProviderByPrefix(String)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProviderManager.html#getProviderByPrefix%28java.lang.String%29) and [EntityProviderManager.getProviderByPrefixAndCapability(String, Class)](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\EntityProviderManager.html#getProviderByPrefixAndCapability%28java.lang.String,%20java.lang.Class%29) among others, you can cache the prefix name and capabilities if you need to

Parameters:

provider - the currently registered provider that you are being notified about

Interface: Formats

public interface Formats

Defines a list of possible format types (extensions) which can be handled and indicates which are handled internally

Methods:

### findEntityRefs

java.util.List<java.lang.String> findEntityRefs(java.lang.String[] prefixes,

java.lang.String[] name,

java.lang.String[] searchValue,

boolean exactMatch)

Allows searching for entities by meta property values, at least one of the params (prefix, name, searchValue) must be set in order to do a search, (searches which return all references to all entities with properties are not allowed)   
WARNING: this search is very fast but will not actually limit by properties that are placed on the entity itself or return the entity itself and is not a substitute for an API which allows searches of your entities (e.g List getYourStuff(Search search); )

Parameters:

prefixes - limit the search to a specific entity prefix or set of prefixes, this must be set and cannot be an empty array

name - limit the property names to search for, can be null to return all names

searchValue - limit the search by property values can be null to return all values, must be the same size as the name array if it is not null, (i.e. this cannot be set without setting at least one name)

exactMatch - if true then only match property values exactly, otherwise use a "like" search

Returns:

a list of entity references for all entities matching the search

### getProperties

java.util.Map<java.lang.String,java.lang.String> getProperties(java.lang.String reference)

Retrieve all meta properties for this entity as a map of name->value

Parameters:

reference - a globally unique reference to an entity

Returns:

a map of String (name) -> String (value)

### getPropertyValue

java.lang.String getPropertyValue(java.lang.String reference,

java.lang.String name)

Retrieve a meta property value for a specific property name on a specific entity

Parameters:

reference - a globally unique reference to an entity

name - the name (key) for this property

Returns:

the property value for this name and entity, null if none is set

Interface: RequestGetter

public interface RequestGetter

Allows for getting to the request and response objects for the current thread

Methods:

### getRequest

javax.servlet.http.HttpServletRequest getRequest()

Returns:

the current request for this thread or null if none can be found

### getResponse

javax.servlet.http.HttpServletResponse getResponse()

Returns:

the current response for this thread or null if none can be found

Interface: RequestGetterWrite

public interface RequestGetterWrite

extends [RequestGetter](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestGetter.html)

Allows for getting to the request and response objects for the current thread

Methods:

### setRequest

void setRequest(javax.servlet.http.HttpServletRequest req)

Sets the request for the current thread, this will be cleared when the thread closes

Parameters:

req - the current HttpServletRequest

### setResponse

void setResponse(javax.servlet.http.HttpServletResponse res)

Sets the response for the current thread, this will be closed when the thread closes

Parameters:

res - the current HttpServletResponse

Interface: RequestStorage

public interface RequestStorage

This allows access to values which are stored in the current request thread, these values are inaccessible outside of a request and will be destroyed when the thread ends  
This also "magically" exposes all the values in the request (attributes and params) as if they were stored in the map as well, if there are conflicts then locally stored data always wins over data from the request  
Standard reserved keys have values that are always available:

* \_locale : Locale
* \_requestEntityReference : String
* \_requestActive : [true,false]
* \_requestOrigin : ['REST','EXTERNAL','INTERNAL']

Methods:

### getStorageMapCopy

java.util.Map<java.lang.String,java.lang.Object> getStorageMapCopy()

Get the data as a map for easy access to the full set of keys/values, this is a copy and changing it has no effect on the data in the request

Returns:

a copy of the internal storage of request keys and values as a map, may be empty but will not be null

### getStorageMapCopy

java.util.Map<java.lang.String,java.lang.Object> getStorageMapCopy(boolean includeInternal,

boolean includeHeaders,

boolean includeParams,

boolean includeAttributes)

Special version which allows getting only the parts that are desired

Parameters:

includeInternal - include the internal request values

includeHeaders - include the request headers

includeParams - include the request parameters

includeAttributes - include the request attributes

Returns:

the map with the requested values

### getStoredValue

java.lang.Object getStoredValue(java.lang.String key)

Get a value that is stored in the request for a specific key

Parameters:

key - a key for a stored value

Returns:

the stored value if found OR null if not found

Throws:

java.lang.IllegalArgumentException - if the key is null

### getStoredValueAsType

<T> T getStoredValueAsType(java.lang.Class<T> type,

java.lang.String key)

Type Parameters:

T -

Parameters:

type - an object type to attempt to convert the object to

key - a key for the stored value

Returns:

the stored value converted to the requested type OR null if none found

Throws:

java.lang.IllegalArgumentException - if the type or key is null

java.util.IllegalFormatConversionException - if the conversion cannot be completed

Interface: RequestStorageWrite

public interface RequestStorageWrite

extends [RequestStorage](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestStorage.html)

This allows write access to values which are stored in the current request thread, these values are inaccessible outside of a request and will be destroyed when the thread ends  
This also "magically" exposes all the values in the request (attributes and params) as if they were stored in the map as well, if there are conflicts then locally stored data always wins over data from the request  
Standard reserved keys have values that are always available:

* \_locale : Locale
* \_requestEntityReference : String
* \_requestActive : [true,false]
* \_requestOrigin : ['REST','EXTERNAL','INTERNAL']

Methods:

### reset

void reset()

Clear all values in the request storage (does not wipe the values form the request itself)

### setRequestValue

void setRequestValue(java.lang.String key,

java.lang.Object value)

Allows user to set the value of a key directly, including reserved keys

Parameters:

key - the name of the value

value - the value to store

Throws:

java.lang.IllegalArgumentException - if the key OR value are null, also if an attempt is made to change a reserved value (see [RequestStorageWrite](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestStorageWrite.html))

### setRequestValues

void setRequestValues(java.util.Map<java.lang.String,java.lang.Object> params)

Place all these params into the request storage

Parameters:

params - map of string -> value params

Throws:

java.lang.IllegalArgumentException - if the key OR value are null, also if an attempt is made to change a reserved value (see [RequestStorageWrite](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestStorageWrite.html))

### setStoredValue

void setStoredValue(java.lang.String key,

java.lang.Object value)

Store a value in the request storage with an associated key

Parameters:

key - a key for a stored value

value - an object to store

Throws:

java.lang.IllegalArgumentException - if the key OR value are null, also if an attempt is made to change a reserved value (see [RequestStorageWrite](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestStorageWrite.html))

Interface: SearchProvider

public interface SearchProvider

Defines the methods related to searching for entities (shared between interfaces), implementing this allows for support for the core EB search functionality

Methods:

### add

boolean add(java.lang.String reference,

[SearchContent](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\SearchContent.html) content)

Add some content to the search index for a given reference, this will replace existing content (not merge)

Parameters:

reference - a globally unique reference to an entity (e.g. /myprefix/myid), consists of the entity prefix and optional segments (normally the id at least)

content - [OPTIONAL] if null the content will be retrieved by called the [Searchable](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Searchable.html) methods, otherwise this will simply index the content it is given into the search system

Returns:

true if added a new search index entry, or false if it replaced an existing one

Throws:

java.lang.UnsupportedOperationException - if this reference cannot be searched

### remove

boolean remove(java.lang.String reference)

Parameters:

reference - a globally unique reference to an entity (e.g. /myprefix/myid), consists of the entity prefix and optional segments (normally the id at least)

Returns:

true if the index entry was removed, false otherwise

### resetSearchIndexes

void resetSearchIndexes(java.lang.String context)

WARNING: this should mostly never be run but it tells the search provider to purge the indexes and to request new search data for the given context

Parameters:

context - this generally represents either a site or a group in the system

### search

[SearchResults](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\SearchResults.html) search([QuerySearch](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\QuerySearch.html) query)

Do a search for entities and get back the listing of results

Parameters:

query - the search query object (allows paging etc.)

Returns:

the search results

Interface: TagProvider

public interface TagProvider

Defines the methods related to tagging entities (shared between interfaces)

Methods:

### addTagsToEntity

void addTagsToEntity(java.lang.String reference,

java.lang.String[] tags)

Add these tags to the an entity, will have no effect if the entity already has these tags

Parameters:

reference - a globally unique reference to an entity (e.g. /myprefix/myid), consists of the entity prefix and optional segments (normally the id at least)

tags - a set of the tags to associate with this entity

Throws:

java.lang.UnsupportedOperationException - if this reference cannot be tagged

### getTagsForEntity

java.util.List<java.lang.String> getTagsForEntity(java.lang.String reference)

Get the set of tags which are associated with this entity

Parameters:

reference - a globally unique reference to an entity (e.g. /myprefix/myid), consists of the entity prefix and optional segments (normally the id at least)

Returns:

a set of the tags which are associated with this entity

Throws:

java.lang.UnsupportedOperationException - if this reference cannot be tagged

### removeTagsFromEntity

void removeTagsFromEntity(java.lang.String reference,

java.lang.String[] tags)

Removes these tags from this entity, will have no effect if the tags do not exist on this entity

Parameters:

reference - a globally unique reference to an entity (e.g. /myprefix/myid), consists of the entity prefix and optional segments (normally the id at least)

tags - a set of the tags to remove from this entity, if this is empty then nothing happens, if this includes tags that do not exist on this entity then they are ignored

Throws:

java.lang.UnsupportedOperationException - if this reference cannot be tagged

### setTagsForEntity

void setTagsForEntity(java.lang.String reference,

java.lang.String[] tags)

Sets the tags which are associated with this entity, this overwrites any current tags and makes the input tags the only current tags for this entity

Parameters:

reference - a globally unique reference to an entity (e.g. /myprefix/myid), consists of the entity prefix and optional segments (normally the id at least)

tags - a set of the tags to associate with this entity, setting this to an empty set will remove all tags from this entity

Throws:

java.lang.UnsupportedOperationException - if this reference cannot be tagged

Interface: TagSearchService

public interface TagSearchService

extends [TagProvider](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\TagProvider.html)

Defines the methods necessary for searching for entities by tags (shared interface)

Methods:

### findEntitesByTags

java.util.List<[EntityData](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\extension\EntityData.html)> findEntitesByTags(java.lang.String[] tags,

java.lang.String[] prefixes,

boolean matchAll,

[Search](file:///D:\newly%20inserted\lms\docx\eb-api\doc\org\sakaiproject\entitybroker\entityprovider\search\Search.html) search)

Search for all entities with a set of tags, prefixes, and a search

Parameters:

tags - a set of tags defined on these entities which should match

prefixes - (optional) limit the search to a given set of prefixes

matchAll - if true then all tags must be matched, else find entities with any tags in the set given

search - (optional) search params which are used to limit the return (paging and max/limit)

Returns:

a list of entity data objects representing all found entities

Package org.sakaiproject.entitybroker.entityprovider.search

public class **Search**

extends java.lang.Object

This is a simple class which allows the passing of a set of search parameters in a nice way  
Example usage:  
Search s1 = new Search("title", curTitle); // search where title equals value of curTitle  
Search s2 = new Search("title", curTitle, Restriction.NOT\_EQUALS); // search where title not equals value of curTitle  
Search s2 = new Search(  
new Restriction("title", curTitle),  
new Order("title")  
); // search where title equals value of curTitle and order is by title ascending

**Package org.sakaiproject.entitybroker.exception**

public class **EntityBrokerException**

extends java.lang.RuntimeException

The base class for all exceptions out of the entity broker system

public class **EntityException**

extends [EntityBrokerException](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\exception\EntityBrokerException.html)

This exception is used to indicate that a problem occurred attempting to get an entity, the reference to the failed entity should be returned in the exception along with the response code which is indicative of the failure (if possible) HTTP Response Codes (http://en.wikipedia.org/wiki/List\_of\_HTTP\_status\_codes)  
**Client Error**  
400 Bad Request The request could not be understood by the server due to malformed syntax.   
401 Unauthorized The request requires user authentication. This will cause a SecurityException to be thrown in the direct servlet.   
404 Not Found The requested resource could not be found. Typically indicates the entity as identified cannot be located.   
405 Method Not Allowed The method specified in the Request-Line is not allowed for the resource identified by the Request-URI. The response MUST include an Allow header containing a list of valid methods for the requested resource.   
422 Unprocessable Entity The server understands the media type of the request entity, but was unable to process the contained instructions.   
501 Not Implemented The prefix indicated is not handled by the entity system  
  
**Redirection**  
303 See Other The response to the request can be found under a different URI and SHOULD be retrieved using a GET method on that resource. This method exists primarily to allow the output of a POST-activated script to redirect the user agent to a selected resource.   
  
**Successful**  
200 OK The request has succeeded.  
201 Created The request has been fulfilled and resulted in a new resource being created. The newly created resource can be referenced by the URI(s) returned in the entity of the response, with the most specific URI for the resource given by a Location header field.   
204 No Content The server has fulfilled the request but does not need to return an entity-body, and might want to return updated metainformation.

**Package org.sakaiproject.entitybroker.providers**

public interface **EntityRESTProvider**

Handles anything REST based that is not part of the core EB registration piece

public interface **EntityPropertiesService**

This provides the entity system with a way to access properties (and then exposes this to others via the developer helper service)

public class **UserEntityProvider**

extends [AbstractEntityProvider](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\util\AbstractEntityProvider.html)

implements [CoreEntityProvider](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html), [RESTful](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\RESTful.html), [Describeable](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\entityprovider\capabilities\Describeable.html)

Entity Provider for users

**Package org.sakaiproject.entitybroker.util**

public class **EntityResponse**

extends java.lang.Object

This is here to contain the information we get back from an entity (http) request fired by the[EntityBroker.fireEntityRequest(String, String, String, Map, Object)](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\EntityBroker.html#fireEntityRequest(java.lang.String, java.lang.String, java.lang.String, java.util.Map, java.lang.Object)) method  
This object and the data in it should be considered immutable

**entitybroker-impl**

Package org.sakaiproject.entitybroker.dao

public interface EntityBrokerDao

extends org.sakaiproject.genericdao.api.GeneralGenericDao

Interface for internal proxy only

public class EntityTagApplication

extends java.lang.Object

This is the persistent object for storing entity tag applications, i.e. the combination of an entity and a tag, or the instance of a tag on an entity

Package org.sakaiproject.entitybroker.dao.impl

public class EntityBrokerDaoImpl

extends org.sakaiproject.genericdao.springjdbc.JdbcGeneralGenericDao

implements [EntityBrokerDao](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\dao\EntityBrokerDao.html)

Internal dao for entity broker internal services

Package org.sakaiproject.entitybroker.impl

public class EntityBrokerCoreServiceManager

extends java.lang.Object

This allows easy startup of the core entitybroker services in a way which avoids the developer having to know anything about it, anyone who wants to startup the entitybroker core would create an instance of this class Note that the [DeveloperHelperService](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\DeveloperHelperService.html) has to be started separately

public class EntityBrokerManagerImpl

extends java.lang.Object

implements [EntityBrokerManager](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\EntityBrokerManager.html)

This is the internal service for handling entities, most of the work done by entity broker is handled here  
This should be used in preference to the EntityBroker directly by implementation classes that are part of the EntityBroker system, rather than the user-facing EntityBroker directly.

Package org.sakaiproject.entitybroker.impl.devhelper

public class DeveloperHelperServiceImpl

extends [AbstractDeveloperHelperService](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\util\devhelper\AbstractDeveloperHelperService.html)

implementation of the helper service methods

Package org.sakaiproject.entitybroker.impl.oldentity

public class EntityBrokerEntityProducer

extends java.lang.Object

implements org.sakaiproject.entity.api.EntityProducer

The single standard EntityProducer for all "Entities" managed by the EntityBroker system. Entity objects are only created on demand and will expire when unreferenced.

**entitybroker-mocks:**

Package org.sakaiproject.entitybroker.mocks

public class CoreEntityProviderMock

extends [EntityProviderMock](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\mocks\EntityProviderMock.html)

implements [CoreEntityProvider](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html)

Stub class to make it easier to test things that use an [CoreEntityProvider](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\entityprovider\CoreEntityProvider.html), will perform like the actual class so it can be reliably used for testing

**entitybroker-restimpl**

Package org.sakaiproject.entitybroker.rest

public class EntityActionsManager

extends java.lang.Object

Handles everything related to the custom actions registration and execution

public class EntityBatchHandler

extends java.lang.Object

This handles batch operations internally as much as possible, the idea is to provide for a standard way to reduce huge numbers of calls down to 1 call to the server which puts the data together into a single response

public class EntityBrokerRESTServiceManager

extends java.lang.Object

This creates all the needed services (as if it were a service manager), this will let us create the services we need without too much confusion and ensure we are using the same ones   
If this is used then the services should not be created in some other way but should be initiated here and then exported from this class only

public class EntityDescriptionManager

extends java.lang.Object

This handles all the methods related to generating descriptions for entities, html and xml currently supported

public class EntityRedirectsManager

extends java.lang.Object

Handles everything related the URL redirects handling and processing

public class EntityRESTProviderBase

extends java.lang.Object

implements [EntityRESTProvider](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\providers\EntityRESTProvider.html)

This is the standard entity REST provider which will be created and set in the entity broker manager, it will register itself with the entitybrokermanager on startup or construction, it should also be unregistered correctly

**entitybroker-utils:**

Package org.sakaiproject.entitybroker.util

public abstract class AbstractEntityProvider

extends java.lang.Object

implements [EntityProvider](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html), org.springframework.beans.factory.InitializingBean, org.springframework.beans.factory.DisposableBean

WARNING: Requires Spring 1.2.8 or newer libraries in the classpath   
Makes it easier to write [EntityProvider](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\entityprovider\EntityProvider.html)s in webapps   
A class to extend that gets rid of some of the redundant code that has to be written over and over, causes this provider to be registered when it is created and unregistered when it is destroyed, also includes the [DeveloperHelperService](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\DeveloperHelperService.html) as a protected variable, pairs with the parent bean (org.sakaiproject.entitybroker.entityprovider.AbstractEntityProvider)  
Create your spring bean like so (class is your provider, set whatever properties you are using):

<bean parent="org.sakaiproject.entitybroker.entityprovider.AbstractEntityProvider"

class="org.sakaiproject.entitybroker.entitywebapp.WebappEntityProvider">

<property name="dao" ref="MemoryDao" />

</bean>

Package org.sakaiproject.entitybroker.util.access

public class AccessProviderManagerImpl<T>

extends java.lang.Object

A common generic implementation class for managers of different kinds of access providers.

Package org.sakaiproject.entitybroker.util.request

public class RequestGetterImpl

extends java.lang.Object

implements [RequestGetter](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestGetter.html), [RequestGetterWrite](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestGetterWrite.html)

Service which will retrieve the current request information if it is available, this allows an application scoped bean to get access to request scoped information

public class RequestStorageImpl

extends java.lang.Object

implements [RequestStorageWrite](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\entityprovider\extension\RequestStorageWrite.html)

Impl for the request store, will store values in the request itself and will maintain a map of all request values in a threadlocal which should always be cleared at the end of the request

Package org.sakaiproject.entitybroker.util.servlet

public abstract class DirectServlet

extends javax.servlet.http.HttpServlet

This is the core abstract DirectServlet class which is meant to extended, extend this to plugin whatever system you have for initiating/retrieving the EB services and for handling logins   
  
Direct servlet allows unfettered access to entity URLs within the EB system, it also can handle authentication (login) if required (without breaking an entity URL)

Package org.sakaiproject.entitybroker.util.spring

public class EntityPropertiesServiceSpringImpl

extends [AbstractEntityPropertiesService](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\util\core\AbstractEntityPropertiesService.html)

implements [EntityPropertiesService](file:///C:\Users\administrator\workspacefreeware\entitybroker\doc\org\sakaiproject\entitybroker\providers\EntityPropertiesService.html)

This service allows us to track all the properties files related to describing the capabilities of our entities and the entities themselves, it allows lookup of strings as well

public class ResourceFinder

extends java.lang.Object

Takes a path or list of paths to resources and turns them into different things (file/IS/resource), this also allows us to look on a relative or absolute path and will automatically check the typical places one might expect to put sakai config files  
Checks the environmental and file paths first and then defaults to checking the classloaders  
Allows us to find resources in our pack since the Sakai context classloader is wrong, too bad it is not correct, that would be cool, but it is wrong and it is not cool