

[Home](#) [Products](#) [Services](#) [Partners](#) [Documentation](#) [Community](#) [Downloads](#) [About Us](#)

[Marketplace](#)

COMMUNITY

Welcome

[Dashboard](#)
[Download Now](#)
[Start Here](#)
[Participate](#)
[Contribute](#)

Liferay Projects

[Alloy UI](#)
[Liferay Faces](#)
[Liferay IDE](#)
[Liferay Portal](#)
[Liferay Social Office](#)

Community Projects

Forums

Blogs

Wiki

User Groups

Wiki

Search

[Main](#) | [Proposals](#)

 [FrontPage](#) |  [Recent Changes](#) |  [All Pages](#) |  [Orphan Pages](#) | 
[Draft Pages](#)

JSON Service API

[« Back to Service Builder](#)

Tags:  [service builder](#)  [liferay v5.2](#)  [liferay v5.3](#)  [Details](#)  [web services](#)  [Print](#)

Table of Contents [-]

1. [Introduction](#)
2. [How to generate the JSON API](#)
 - I. [Known limitations](#)
3. [Invoking the JSON API](#)

Special Activities

Security Team
Translation Team
100 Papercuts
BugSquad
Community Verifier
Radio Liferay
2012 Contributor Awards

Upcoming Events

Liferay University

I. [From JavaScript](#)

I. [Resolving parameter type disambiguations](#)

II. [From any other language](#)

4. [Example](#)

Introduction

This article describes the API provided by all Liferay services that allows invoking its methods directly through HTTP using JSON as a data serialization mechanisms. This API is automatically generated by [Service Builder](#), which means that you can also provide it for your custom services developed in the extension environment with no extra effort.

How to generate the JSON API

The JSON API is generated automatically from the remote service interface (for example, from ReportsEntryService) when "ant build-service" is executed for any given service.

Besides, a JavaScript file which contains a proxy for all the methods in that service is generated. This JavaScript file can be found in:

- (Portal Core) portal-web/docroot/html/js/liferay/service.js
- (Ext) extl-web/docroot/html/js/liferay/ext_service.js

Known limitations

Not all Java types can be easily converted to and from JSON. Currently ServiceBuilder supports all native types and some object types. If a given method in the remote service does not get registered in service.js that means that some of its method parameter types were not supported.

Look for and open the file bad_json_types.txt within the sources of your specific version of Liferay to find out which types are not supported.

Invoking the JSON API

From JavaScript

Using the JSON Service API from a portlet using JavaScript cannot be any easier. As mentioned earlier, Service Builder automatically creates a file that encapsulates the invocation of the API methods through AJAX and makes it easy to process the response.

In order to be able to use the API the generated service.js file has to be included whenever the portlet is used. This can be done through regular HTML:

```
<script src="/html/js/liferay/service.js"
language="JavaScript"> </script>
```

or for the extension environment:

```
<script src="/html/js/liferay/ext_service.js"
language="JavaScript"> </script>
```

Or configured in liferay-portlet(-ext).xml so that it's added to the head of the HTML page.

Once the JavaScript file is available the methods of the service can be invoked using the following syntax:

```
Liferay.Service.NAMESPACE.ENTITY.METHOD(
    {
        PARAMETER_1: VALUE_1,
        PARAMETER_2: VALUE_2
    },
    function(message) {
        var exception = message.exception;
```

```
        if (!exception) {  
            // Process Success  
        }  
        else {  
            // Process Exception  
        }  
    }  
}
```

NAMESPACE is the namespace of the service as specified within the service.xml file.

ENTITY is the name of the entity as specified in the 'entity' elements of the service.xml file. For example, to access the ReportsEntryService the name of the entity is ReportsEntry.

METHOD is the name of the method.

message.exception contains the name of the Java exception that may have been thrown by the service.

Resolving parameter type disambiguations

Since JavaScript is untyped and Java is typed in some cases it's necessary to specify the type of a given parameter to make sure the proper method is invoked.

The JavaScript API allows specifying the Java types of the parameters through an special parameter called *serviceParameterTypes*. The value of this parameter should be a comma separated list of Java types (in 5.2 or earlier) or a JSON array of Java types (in 5.3 and later). Here is an example:

```
var serviceParameterTypes = [  
    'java.lang.String',  
    '[Ljava.lang.String;',
```

```

        'com.liferay.portal.service.ServiceContext'
    ];

    Liferay.Service.Asset.AssetTag.addTag(
        {
            name: tagName,
            properties: [],
            serviceContext: jQuery.toJSON(
                {
                    communityPermissions:
communityPermission,
                    guestPermissions:
guestPermission,
                    scopeGroupId:
themeDisplay.getScopeGroupId()
                }
            ),
            serviceParameterTypes:
jQuery.toJSON(serviceParameterTypes)
        },
        ...
    );

```

- Note: This example is valid for 5.3 and later. For earlier versions use:

```

        serviceParameterTypes:
serviceParameterTypes.join(',')

```

serviceParameterTypes: jQuery.toJSON(serviceParameterTypes)

From any other language

You can use any HTTP client and JSON serializer to access the service.

Example

```
public class ReportsEntryServiceImpl extends
ReportsEntryServiceBaseImpl {
    public void print(String msg) {
        System.out.println("ReportsEntryService: " + msg);
    }
    ....
}
```

This method could be invoked from a JSP using the following method (to make it simpler we are linking to the js file directly):

```
<script src="/html/js/liferay/ext_service.js"
language="JavaScript"> </script>

<script language="JavaScript">
    Liferay.Service.Reports.ReportsEntry.print(
        {
            msg: "Invoking a Liferay service via JavaScript",
        },
        function(message) {
            var exception = message.exception;

            if (!exception) {
                // Process Success
            }
            else {
                // Process Exception
            }
        }
    )
}
```

```
);  
</script>
```

For a much more complete example, it's recommended to look at the Tags Admin and Categories Admin (from 5.3) portlets included with Liferay Portal.

 [0 Attachments](#) | 25762 Views

Average (0 Votes)



▼ Comments

Showing 18 Comments



Jonas Yuan

7/10/09 6:52 AM

Thank you, Jorge.

The following are sample code for polls and voting (JSON services calls).

```
{{{<script type="text/javascript">  
Liferay.Service.Polls = {  
  servicePackage: "com.liferay.portlet.polls.service."  
};
```

```
Liferay.Service.Polls.PollsVote = {  
  serviceClassName: Liferay.Service.Polls.servicePackage + "PollsVote" +  
  Liferay.Service.classNameSuffix,
```

```
  addVote: function(params, callback) {
```

```
params.serviceClassName = this.serviceClassName;
params.serviceMethodName = "addVote";

return Liferay.Service.ajax(params, callback);
}
};

function vote() {
var choices = document.cmspollsfm.choiceld;
var choiceld = 0;

for (var i = 0; i < choices.length; i++) {
if (choices.checked) {
choiceld = choices.value;
}
}

if (choiceld != 0) {
Liferay.Service.Polls.PollsVote.addVote(
{
questionId: $poll.getData(),
choiceld: choiceld
}
);
location.reload(true);
}
}

</script>}}}
```



Faisal K

7/13/09 5:21 AM

Hi, It is not related to your Comments, I tried to implement my own JSON Class by extending JSONAction but if user is guest then my JSON Class is not invoking, I know it is not enough to give a solution please check this messageboard entry 'http://www.liferay.com/c/message_boards/find_message?messageId=3516435'



Alexey Kakunin

9/20/09 7:33 AM

Very nice feature - but just a note - at least in Liferay 5.2.3 JSON interface is not generated for services in case, you are doing development in plugin environment



Peter Fox

10/9/09 3:23 AM

How to get the response off a service call?

Imagine that service method call returns a List of objects...

If I do a call like:

```
Liferay.Service.Reports.ReportsEntry.getReports(  
{  
  msg: "Invoking a Liferay service via JavaScript --->Reports",  
},  
{  
  inst: 'myTest',  
},  
function(message) {  
  alert(thi);  
}  
);
```

(not fetching from DB...I create the list is hardcoded on service method and only name it's initialized...just for testing)

As the parameter on call is {inst: "myTest"} it returns a response like:

```
var  
myTest=[{"name":"AAAAA","userId":"","userName":"","companyId":"","uuid":"","createDate":"","modifiedDate":"","entryId":""}, {"name":"BBBBB","userId":"","userName":"","companyId":"","uuid":"","createDate":"","modifiedDate":"","entryId":""}, {"name":"CCCCC","userId":"","userName":"","companyId":"","uuid":"","createDate":"","modifiedDate":"","entryId":""}];
```

My question is how to get this result??? and how to use a callback to get the same result... I'm not able to get it... javascript is not my strong...

I'll be very happy if anyone could help me is this one...

BR



Peter Fox

10/9/09 4:11 AM

Can you help me on how to get JSON service reponse?

Cheers



Tim Stavenger

4/21/10 11:22 AM

I have found that the message object returned by Liferay.Service calls contains a "returnValue" attribute that contains the return value.

For example:

...

```
function(message) {
```

```
alert(message.returnValue);  
}
```



Akshay Shrigadi

6/1/10 4:54 AM

In continuation to Tim's response.

The returned returnValue can be assigned to any of the javascript variable as

...

```
var myJSONObject = message.returnValue;
```

If you still get the text value instead of the actual object, try

```
eval(myJSONObject);
```

Hope this helps!



Yogesh Agrawal

10/20/10 5:29 AM

Is JSON Services are available in GA3 pulgin-portlets ?

**serviceHTTP.java is missing in portlet service..*



Jakub Liska

11/28/10 5:27 PM

Is this possible to do outside EXT ? Cause I just tried and it didn't go well :

https://www.liferay.com/community/forums/-/message_boards/message/6630792



Jakub Liska

1/6/11 5:46 PM

Guys sorry for asking again, but is there going to be the possibility to call "portlet" remote services ? From plugin environment? Because the request to

"/c/portal/json_service" is handled by Portal context and portal classloader only



Jakub Liska

1/6/11 7:32 PM

Got it, I just took a look at the Liferay javascript, the only problem is that the javascript resulting request is `http://host/my-portlet/json` ...which ends up right at the beginning of request processing cause there is no filter or mapping for it. `portal/json_service` request is processed fine but this one doesn't get into `MainServlet` at all. What the heck is that



John Menke

1/10/11 7:11 PM

Jakub,

could you post a summary of your syntax to call json from portlet? I am trying :

```
<script src="/html/js/service.js" language="JavaScript"> </script>
```

```
<script language="JavaScript">
```

```
Liferay.Service.namespace.entity.method(  
{  
  param: "test"  
},  
function(message) {  
  alert("inside function");  
  var exception = message.exception;  
  
  if (!exception) {  
    alert(message.resultValue);  
  }  
});
```

```
}  
else {  
    alert("fail");  
}  
}  
);
```

</script>

but the method never gets called

Thanks for any help - as far as I can tell your one of the few people that have this working

John



Yogesh Agrawal

2/8/11 9:23 PM

Missing ArticleServiceHttp.java in Knowledge Base portlet.

Is the file is not generated by Servicebuilder ??

Thanks for any help



Aston Chan

4/4/11 12:14 PM

Does it work if the parameter to the server call is a user defined object? Instead of simple string or any of the primitive types?

I'm getting this on the LR console:

18:42:42,933 ERROR [JSONServiceAction:397] Unsupported parameter type for class
class com.x.ProductServiceUtil, method addProduct, parameter Product,
and type interface com.x.model.Product

The server code has this signature:

```
/**  
 * Method to add a new Product to the Database.  
 */  
public Product addBehavior(com.x.model.Product newProduct, long userId) throws  
SystemException, PortalException {  
  
Product product = productPersistence.create(new  
ProductPK(newProduct.getPageId(), newProduct.getName()));  
...  
}
```

but the newProduct is ALWAYS null.

So, the question is, does it work if one of my server call contains the Product object?
Or does it only work for primitive parameters?

Thanks.



Aston Chan

4/4/11 1:43 PM

Sorry, pasted wrong server code signature. The server code should be :

```
public Product addProduct(com.x.model.Product newProduct, long userId) throws  
SystemException, PortalException {
```

"The Unsupported parameter type for class"... is always shown on the console window.

Thanks.



Sagar A Vyas

5/27/11 10:48 AM

Hi All,

This is really nice post., i tried this and i got succeed to get user object.but her my Questions is i am trying to get screenName of user that i will get inside this function but i could not get it out side of this function because of Asynchronous call. Can any one help me out ?

Please find code as below.

```
var userScreenName = "";
Liferay.Service.Portal.User.getUserById(
{userId:userId},

function(user){
var exception = user.exception;

if (!exception)
{
var userObj = jQuery.parseJSON(jQuery.toJSON(user));
userScreenName = userObj.screenName;
alert(userScreenName + '==== '+userObj.screenName );
}
}
```

```
else
{
// Process Exception
}

}

);

userScreenName = "/web/" + userScreenName + "/home"
alert("after..." + userScreenName);
$(this).find("a").attr("href", userScreenName);
Thanks,
Sagar Vyas
CIGNEX
```



Markk Cline

8/25/11 2:21 PM

Hi Sagar,

If you haven't figured this out your code should look like this...

#1) It's not clear if you defined `userId` somewhere. I used `themeDisplay` directly instead

#2) The returning object `user` is already an object - you don't need jQuery to parse the JSON again (this causes an error)

And of course make sure you have the `service.js` and `jquery` files added in your portlet or portal.


```
var userScreenName = "";

//alert(themeDisplay.getUserId());

Liferay.Service.Portal.User.getUserById(
{userId:themeDisplay.getUserId()},function(user){
var exception = user.exception;

if (!exception)
{

userScreenName = user.screenName;
alert(userScreenName );
}
else
{

alert('ERROR');
}

}

);
```



Etomy Tang

11/16/11 2:22 AM

service impl code : `public void deleteByIds(String[] ids)`

javascript code:

`Liferay.Service.UserMapping.ApplicationSystem.deleteByIds(`

```
{
  deleteIds : ['abc']
},
function(message){
  alert(message.exception);
});
}
}
);
```

if the parameter type is String[], exception is thrown as below:

```
18:20:28,033 ERROR [ClassLoaderProxy:73] java.lang.IllegalArgumentException:
argument type mismatch
java.lang.IllegalArgumentException: argument type mismatch
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
at com.liferay.portal.kernel.util.MethodHandler.invoke(MethodHandler.java:81)
at com.liferay.portal.kernel.util.ClassLoaderProxy.invoke(ClassLoaderProxy.java:67)
at
com.regaltec.rportal.portlet.usermapping.service.ApplicationSystemServiceClp.deleteByIds(ApplicationSystemServiceClp.java:130)
at com.regaltec.rportal.portlet.usermapping.service.ApplicationSystemServiceUtil.deleteByIds(ApplicationSystemServiceUtil.java:64)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at
```

sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
at
com.liferay.portal.action.JSONServiceAction.getJSON(JSONServiceAction.java:124)
at com.liferay.portal.struts.JSONAction.execute(JSONAction.java:49)
at
org.apache.struts.action.RequestProcessor.processActionPerform(RequestProcessor.java:431)
at org.apache.struts.action.RequestProcessor.process(RequestProcessor.java:236)
at
com.liferay.portal.struts.PortalRequestProcessor.process(PortalRequestProcessor.java:153)
at org.apache.struts.action.ActionServlet.process(ActionServlet.java:1196)
at org.apache.struts.action.ActionServlet.doPost(ActionServlet.java:432)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:637)
at com.liferay.portal.servlet.MainServlet.callParentService(MainServlet.java:508)
at com.liferay.portal.servlet.MainServlet.service(MainServlet.java:485)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:126)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)

at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:126)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.servlet.filters.strip.StripFilter.processFilter(StripFilter.java:261)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:123)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.servlet.filters.themepreview.ThemePreviewFilter.processFilter(ThemePreviewFilter.java:88)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:123)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.servlet.filters.gzip.GZipFilter.processFilter(GZipFilter.java:110)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:123)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.servlet.filters.secure.SecureFilter.processFilter(SecureFilter.java:182)

at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:123)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at
org.apache.catalina.core.ApplicationDispatcher.invoke(ApplicationDispatcher.java:646)
at
org.apache.catalina.core.ApplicationDispatcher.processRequest(ApplicationDispatcher.java:436)
at
org.apache.catalina.core.ApplicationDispatcher.doForward(ApplicationDispatcher.java:374)
at org.apache.catalina.core.ApplicationDispatcher.forward(ApplicationDispatcher.java:302)
at com.liferay.portal.servlet.I18nServlet.service(I18nServlet.java:102)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.servlet.filters.sso.ntlm.NtlmPostFilter.processFilter(NtlmPostFilter.java:81)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:123)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)

at com.liferay.portal.sharepoint.SharepointFilter.processFilter(SharepointFilter.java:179)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:123)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.servlet.filters.virtualhost.VirtualHostFilter.processFilter(VirtualHostFilter.java:240)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:123)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:126)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:126)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at org.tuckey.web.filters.urlrewrite.UrlRewriteFilter.doFilter(UrlRewriteFilter.java:738)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)

java:206)
at com.liferay.portal.kernel.servlet.BaseFilter.processFilter(BaseFilter.java:196)
at com.liferay.portal.servlet.filters.threadlocal.ThreadLocalFilter.processFilter(ThreadLocalFilter.java:35)
at com.liferay.portal.kernel.servlet.BaseFilter.doFilter(BaseFilter.java:123)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
at
org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:233)
at
org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:191)
at
org.apache.catalina.authenticator.AuthenticatorBase.invoke(AuthenticatorBase.java:470)
at
org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:127)
at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
at
org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)
)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:298)
at org.apache.coyote.http11.Http11Processor.process(Http11Processor.java:857)
at
org.apache.coyote.http11.Http11Protocol\$Http11ConnectionHandler.process(Http11Protocol.java:588)
at org.apache.tomcat.util.net.JIoEndpoint\$Worker.run(JIoEndpoint.java:489)
at java.lang.Thread.run(Thread.java:619)

if i change the paramter type for long[], it is work right, why?



NEWS



BLOGS



TWITTER



FACEBOOK



LINKEDIN

Powered by Liferay Portal 

[PRIVACY POLICY](#)

[SITEMAP](#)

[CONTACT US](#)

[ABOUT US](#)

[CAREERS](#)

© 2012 LIFERAY INC. ALL RIGHTS RESERVED