

# Pageflex Services

## Storefront Extension | jSINI

---

### Summary

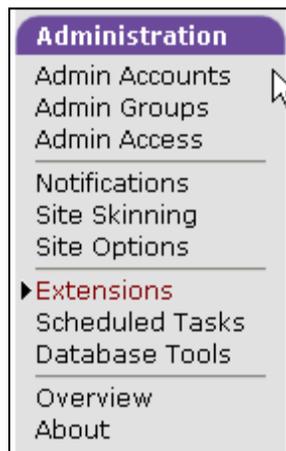
The **jSINI** extension provides access to a subset of the Storefront API via JavaScript. It only exposes API methods that return strings. While implementing API calls in JavaScript may certainly be simpler than writing extension it does expose the API in client side code which has the potential to create some security risks. In jSINI 2.0 we've introduced the ability to limit API access via jSINI and to disallow the use of SetValue calls.

### Configuring jSINI

#### Extension Setup

After installing the extension (by running the setup file **ExtensionManager for jSINI.exe**), follow these steps to configure **jSINI**:

1. Login to the Storefront deployment's administrator site.
2. Select the **Extensions** page in the left-hand panel.



- Click on the extension **Services: Punch Out Integration** to enter the configuration.

<input type="checkbox"/>	<a href="#">Services: Payflow Pro Payment Gateway</a>		OK	<a href="#">Se</a>
<input type="checkbox"/>	<a href="#">Services: Sku Map</a>		OK	<a href="#">Se</a>
<input type="checkbox"/>	<a href="#">Services: jSINI</a>		OK	<a href="#">Se</a>
<input type="checkbox"/>	<a href="#">Services: Canadian Tax</a>			<a href="#">Se</a>
<input type="checkbox"/>	<a href="#">Services: Discount Code</a>			<a href="#">Se</a>

- Set the **Active** property to “Yes”.

### Configure Extension: "Services: jSINI"

Set the properties of this extension.

**General Properties**

Path: Services\SuperInsert.dll

Active:  Yes     No

Services: jSINI - [Open Doc](#) - Up to date.

Limit API Access

Don't allow SetValue calls

- Questions? Feedback? Projects? [Send email to services@pageflex.com](mailto:services@pageflex.com) -  
10.0.0.89

- Checking Limit API Access restricts API calls to objects that are owned by the current logged in user. An attempt to retrieve a value for an object that is not owned by the current logged on user will return undefined and “Illegal access attempt” will be recorded in the Storefront log.
- Checking Don't allow SetValue calls attempts at SetValue calls will return undefined and “Illegal access attempt” will be recorded in the Storefront log.

---

## Usage

**jSINI** has two main purposes:

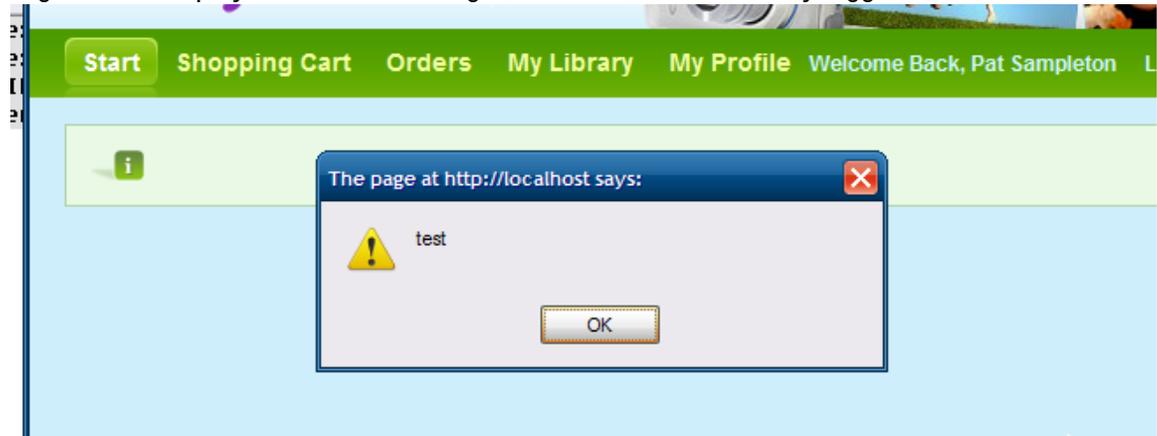
1. Allowing calling of SINI calls from Javascript in Storefront.
2. Allowing injecting custom HTML inserts on almost every page.

### Basic Setup and Proof of Concept

1. Once you have jSINI installed enable it in the extension configuration.
2. Go to the Webpages\Custom\LoadInserts\ directory of your deployment.
3. Create a text file called *usercontentstart.html* and place the following inside it:

```
<script type="text/javascript" src="jSINI.js"></script>
<script type="text/javascript">
    alert(CallSINIMethod("GetValue",
        ["SystemProperty", "LoggedInUserName", null]));
</script>
```

4. Logon to the deployment. You should get an alert with the currently logged on username.



5. That's the basics! Read on for the reference.

jSINI can be used in HTML Literals, Inserts, or anywhere else you can place HTML directly in Storefront Deployment. It only works within the Storefront user site. To use it include the jSINI.js script file:

```
<script type="text/javascript" src="jSINI.js"></script>
```

You can then call CallSINIMethod anywhere in your JavaScript. The first argument is the name of the SINI method you wish to call (see Chapter 6 of the Programmer's Reference). The second is an array of arguments for the SINI method. The following sample gets the documents in the shopping cart of the currently logged in user:

```
var user_id = CallSINIMethod("GetValue",
    ["SystemProperty", "LoggedInUserID", null]);

var docs = CallSINIMethod("GetListValue",
    ["UserListProperty", "DocumentsInShoppingCart",
    user_id]);
```

There are only a handful of SINI methods that currently do not work. Those are any method which returns something other than a string or string array.

## LoadInserts

The jSINI extension allows you to inject custom HTML into the Display Message area on almost every Storefront web page. One powerful implementation of this capability is to use jQuery in your custom HTML to hide the the Display Message area and create custom inserts almost anywhere else on the page. For more information on jQuery see <http://jquery.com>.

Any files that end with a Storefront page name in all lower case with html as a file extension in the Webpages\Custom\LoadInserts\ will be loaded and displayed. It will be displayed in a Display Message dialog. If you were to make a file named my\_usercontentstart.html with the content `<h1>HelloWorld</h1>` you will get:



Another example file name would be `myotherscript_usercontentstart.html` to make another `LoadInsert` on the `UserContentStart.aspx` page.

You can hide the Display Message box with the following JavaScript (in one continuous line):

```
document.getElementById('myScript').
parentElement.parentElement.parentElement.
parentElement.parentElement.parentElement.
parentElement.style.display='none';
```

If you are using jQuery this can be done even more cleanly with the following javascript:

```
$("#myScript").parents(".warningAreaInlineInfo").hide();
```

*Note: Be sure to make a div in your load insert with the id myScript.*

## New in version 2.7 - Support for GetAllValues

Since making multiple JSINI calls causes a round trip (client -server) call, a significant delay can occur in page rendering. This code is intended to reduce the overhead required for each JSINI invocation.

For example if multiple GetValue calls are required to get values for OrderFields you can do the following

```
//  
// Get all the order FieldValues  
var allOrderFieldValues = CallJSINIMethod("GetAllValues", ["OrderField", orderId]);
```

Returns a string in the following tokenized format to a local variable:

Field1 Name | Field1 Value | Object1 Id, Field2 Name | Field2 Value | Object2 Id

```
//  
// Get price of the order before adjustment  
var price = getFieldValue(allOrderFieldValues, "PaymentOrderAmount");
```

To access the field value of the specific field returned in the array, you can use the following method

```
function getFieldValue(data, fieldName) {  
    var tokens = data.split(",");  
    for (var index = 0; index < tokens.length; index++) {  
        var tokenData = tokens[index].split("|");  
        if (tokenData[0] == fieldName) {  
            return tokenData[1];  
        }  
    }  
    return null;  
}
```

**NOTE:** If the field value is empty then a "NULL" token will be returned in the string

---

## Version History

### 2.7.3 – March 7, 2013

- Enabled CreateDocument() calls

### 2.7.1 – September 10, 2012

- Documentation fix

### 2.7.0 – July 30, 2012

- Added code to support GetAllValues call that returns tokenized string. This method can reduce overhead of making multiple GetValue calls, thus decreasing page rendering time.

### 2.6.0 – July 26, 2012

- Added the localization files – for support of other languages

### 2.5.0 – May 30, 2012

- Add option(s) to restrict API access.

### 1.5.0 – Jan 17, 2012

- Compiled for SF 8.0

### 1.4.1 – May 9, 2011

- Updated to SF 7.7

- 

### 1.4.0 – July 13, 2010

- Added escape function to arguments passed to ASPX page.

### 1.2.3 – June 10, 2009

- Needed to incorporate override version of extension manager.

### 1.2.2 – May 19, 2009

- Improved Load Insert section of the documentation.

### 1.2.1 – April 30, 2009

- CallSINIMethod now returns an array if the method is GetListValue. Otherwise, it returns a string.

### 1.2 – April 27, 2009

- The extension now returns an XML block to the browser, which should fix problems with characters needing to be encoded.

### 1.1 – April 22, 2009

- Made it possible to have multiple load inserts on a single page by just checking endswith.
- Put in Version checking config page.

### 1.0.2 – March 27, 2009

- Fixed a bug where Firefox was returning “Length Required”.
- Fixed a bug where IE was caching results.

### 1.0.1 – March 6, 2009

- Fixed a bug to handle null return values.

### 1.0 – February 20, 2009

- First Release

---

## Pageflex Storefront Extension Installation Guide

Information in this document is subject to change without notice.

NEITHER PAGEFLEX NOR MSDH INC. MAKES ANY WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Neither Pageflex nor MSDH Inc. shall be liable for errors herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

© Copyright 2008 MSDH Inc. and Pageflex, Marlborough, MA. All rights reserved. No part of this document may be photocopied, reproduced, or translated without the prior written consent of Pageflex.

MSDH Inc., 500 Nickerson Road, Marlborough, MA 01752

Pageflex, a division of MSDH, 500 Nickerson Road, Marlborough, MA 01752

Worldwide phone number: 1-617-520-8600

Phone number in the U.S. and Canada: 800-551-0549

Internet: <http://www.pageflex.com>

Printed and bound in the United States of America.

9/10/2012