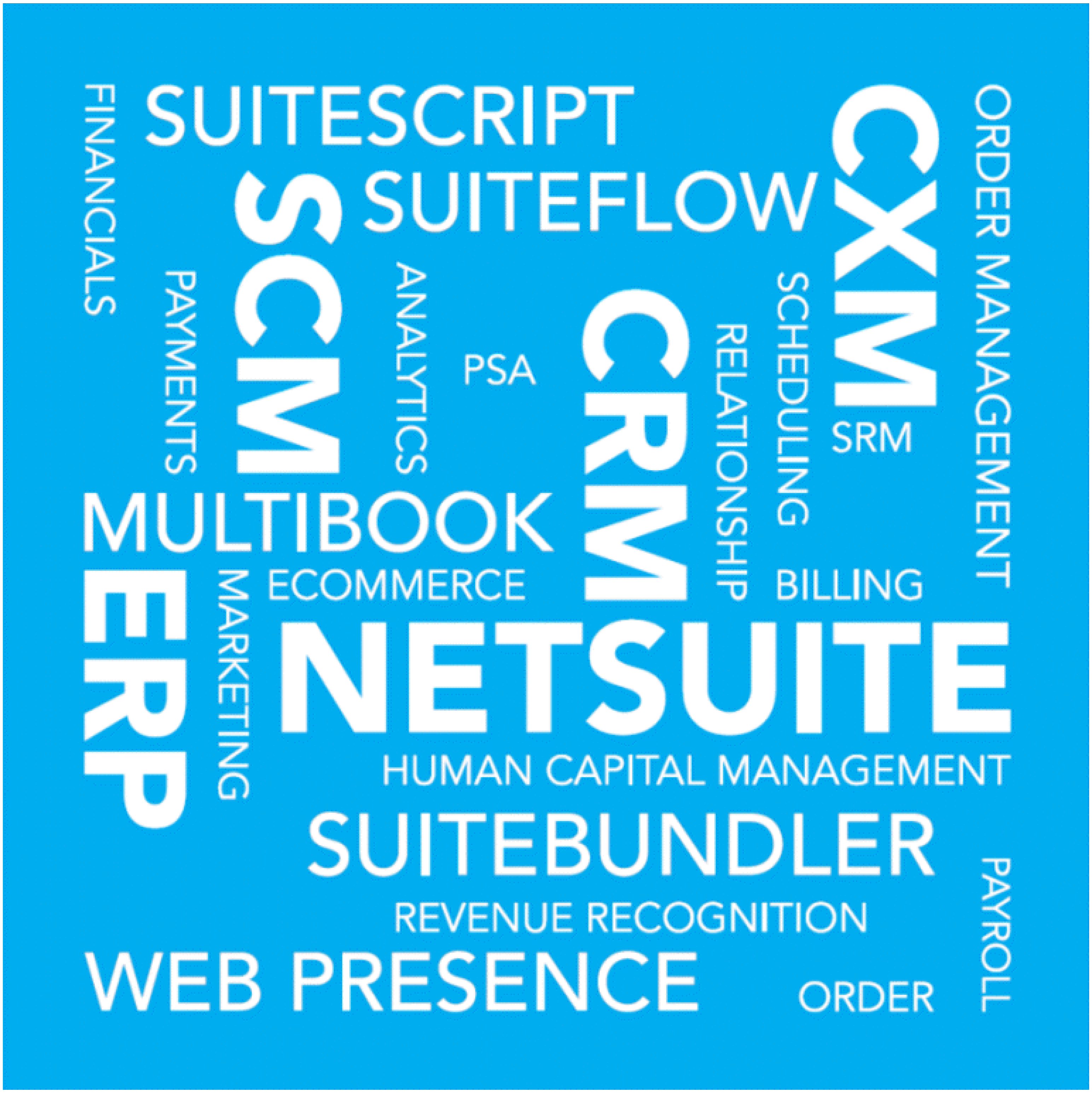
Introduction to RESTlets



May 10, 2017 2017.1

General Notices

Table of Contents

RESTlets

You can deploy server-side scripts that interact with NetSuite data following RESTful principles. RESTlets extend the SuiteScript API to allow custom integrations with NetSuite. Some benefits of using RESTlets include the ability to:

- Find opportunities to enhance usability and performance, by implementing a RESTful integration that is more lightweight and flexible than SOAP-based web services.
- Support stateless communication between client and server.
- Control client and server implementation.
- Use built-in authentication based on token or user credentials in the HTTP header.
- Develop mobile clients on platforms such as iPhone and Android.
- Integrate external Web-based applications such as Gmail or Google Apps.
- Create backends for Suitelet-based user interfaces.

RESTlets offer ease of adoption for developers familiar with SuiteScript and support more behaviors than NetSuite's SOAP-based web services, which are limited to those defined as SuiteTalk operations. RESTlets are also more secure than Suitelets, which are made available to users without login. For a more detailed comparison, see RESTlets vs. Other NetSuite Integration Options.

For more information about the REST architectural style, a description of the constraints and principles is available at http://en.wikipedia.org/wiki/Representational_State_Transfer.

For details about working with RESTlets, see:

- Working with RESTlets
- RESTlets vs. Other NetSuite Integration Options
- Creating a RESTlet
- Debugging a RESTlet
- Sample RESTlet Code
- Sample RESTlet Input Formats
- RESTlet Status Codes and Error Message Formats
- Tracking and Managing RESTlet Activity

Important: To maintain the highest security standards for connectivity, NetSuite periodically updates server certificates and the trusted certificate store we use for https requests. We expect client applications that integrate with NetSuite to use an up-to-date certificate store, to ensure that integrations do not break due to certificates expiring or changing security standards. This issue typically does not affect modern browsers because these clients are maintained to use the latest certificates.

Working with RESTlets

You can invoke a RESTlet via an HTTP request to a system-generated URL. RESTlets send and receive content in a request-response model using HTTP verbs, HTTP headers, HTTP status codes, URLs, and standard data formats.

RESTlets support the entire SuiteScript API and general SuiteScript features such as debugging. For general information about using SuiteScript, see the help topic SuiteScript - The Basics.

The following topics provide details specific to the RESTlet type of script:

- RESTlet Script Execution
- Authentication for RESTlets
- RESTlet URL and Domain
- Using the REST roles Service to Get User Accounts, Roles, and Domains
- Supported Input and Output Content Types for RESTlets
- Supported Functions for RESTlets
- RESTlet Governance and Session Management
- RESTlet Debugging
- RESTlet Error Handling
- RESTlet Security

RESTlet Script Execution

The following steps provide an overview of the RESTlet execution process:

1. A client initiates an HTTP request for a system-generated URL. This request can come from an external client or from a client hosted by NetSuite.
2. The sender of the HTTP request is authenticated either through request-level credentials passed by the NetSuite-specific method NLAuth, the NetSuite implementation of OAuth 1.0, or through a check for an existing NetSuite session.
   - For an externally hosted client using NLAuth, request-level credentials are required.
   - For an externally hosted client using OAuth 1.0, a token is required.
   - For a NetSuite-hosted client, the existing NetSuite session is reused.

   See Authentication for RESTlets.
3. The RESTlet script is invoked, providing access to the server SuiteScript API.
4. A string, or an object that has been deserialized according to a predefined specification, is passed in to the RESTlet. See Supported Input and Output Content Types for RESTlets.
5. The RESTlet interrogates the string or object, and can perform any of the full range of SuiteScript actions.
6. The RESTlet returns results as a string or a serialized object. nlobjResponse

Important: The URL used to invoke a RESTlet varies according to whether the RESTlet client is externally hosted or hosted by NetSuite. See RESTlet URL and Domain.

Authentication for RESTlets

RESTlets require authentication and calls are processed synchronously. The way to provide login credentials for a RESTlet varies according to whether the RESTlet is called from an external client or from a client hosted by NetSuite, such as a client SuiteScript.

Important: RESTlets must use rest URLs to connect to NetSuite. If the RESTlet call comes from an external client, the URL must include a domain specific to your NetSuite account. This domain can change without notice. For that reason, you must dynamically discover the correct domain when calling RESTlets from an external client. To handle this task, use the roles service, as described in Using the REST roles Service to Get User Accounts, Roles, and Domains.

- For a RESTlet called from an external client, you can use OAuth or the NetSuite-specific method **NLAuth** in the HTTP Authorization header. OAuth uses token-based authentication (TBA) to access resources on behalf of a user, eliminating the need to share login credentials such as username and password. See Using OAuth in the Authorization Header.

  NLAuth passes in NetSuite login credentials such as company ID, user name, password, and role. See Using NLAuth in the Authorization Header.
- For a RESTlet called from a client hosted by the same NetSuite account that hosts the RESTlet, you do not need to pass authentication information in the HTTP request. A check for all valid NetSuite session cookies occurs, and this existing session is reused.

Important: RESTlet authentication can use either the HTTP Authorization header or all session cookies, but not both. Please ensure that your script uses only one form of authentication.

Setting up Token-based Authentication for a RESTlet integration

NetSuite supports token-based authentication (TBA) a robust, industry standard-based mechanism that increases the overall security of the system. This authentication mechanism enables client applications to use a token to access NetSuite through APIs, eliminating the need for RESTlets to store user credentials. A token is valid for one specific company, user entity, and role only.

Note: Web Services Only roles are only for access to NetSuite through web services. Roles with the Web Services Only restriction will not work with RESTlets.

For more information, see the help topic Getting Started with Token-based Authentication.

When you use token-based authentication, password rotation policies in the account do not apply to tokens and password management is unnecessary for your RESTlets integrations. Token-based authentication allows integrations to comply with any authentication policy that is deployed in a NetSuite account for UI login, such as SAML Single Sign-on, Inbound Single Sign-on, or Two-Factor Authentication. To enable token-based authentication, see the help topic Enabling the Token-based Authentication Feature.

You can create a token and assign it to a user by logging in to NetSuite as an administrator and generating token credentials manually. NetSuite users can also generate token for themselves. See the help topic Token-based Authentication Permissions.

For code snippets and examples of signature creation and token-based authentication, see SuiteAnswer 42171 and SuiteAnswer 42019.

For information about calling a token endpoint to issue or revoke a token, see the help topic TBA: Calling a Token Endpoint in the Token-based Authentication section of the Help Center.

Using OAuth in the Authorization Header

When calling a RESTlet, follow the OAuth 1.0 specification to generate a token. A description of the OAuth 1.0 protocol and signature validation is available at https://tools.ietf.org/html/rfc5849.

Note: Supported Signature methods are `HMAC-SHA1` and `HMAC-SHA256`.

To obtain the parameter values required for OAuth, refer to the confirmation page for your NetSuite token-based application. See the help topic Creating Applications for Token-based Authentication.

OAuth passes in the following parameters:

- `oauth_signature` (required) - Credentials to verify the authenticity of the request, generated by calling your application. The Token Secret and Consumer Secret are constructed as a key to sign the request, using a supported signature method (HMAC-SHA1 or HMAC-SHA256).
- `oauth_version` (optional) - Must be set to “1.0”.
- `oauth_nonce` (required) - Passes in a unique, random, alphanumeric string. String must be a minimum of 6 characters, and the maximum length is 64 characters. Used to verify that a request has never been made before.
- `oauth_signature_method` (required) - Must be set to HMAC-SHA1 or HMAC-SHA256. Declares which signature method is used.
- `oauth_consumer_key` (required) - Consumer Key (client application ID) generated for the token-based application in NetSuite. The unique value is matched to the token to establish ownership of the token.
- `oauth_token` (required) - Token ID generated for the token-based application in NetSuite.
- `oauth_timestamp` (required) - Passes in a positive integer expressed as the number of seconds since January 1, 1970 GMT.
- `realm` (required) - NetSuite company ID

To map error codes to a parameter issue, see Notes about RESTlet Errors.

Note: NetSuite recommends entering the key value pairs without leading or trailing spaces (`realm="000068"` not `realm=  "000068"`) as shown in the following example.

Example RESTlet request using OAuth to access a protected resource

```
GET https://rest.na1.netsuite.com/app/site/hosting/restlet.nl?script=1&deploy=1 HTTP/1.1
Authorization:
               OAuth oauth_signature="MgN1gZztYspNQXA576plPD14OWM%3D",
               oauth_version="1.0",
               oauth_nonce="207310548",
               oauth_signature_method="HMAC-SHA1",
               oauth_consumer_key="fvFwnmvurChjol7SZiF2pQ1oJ%2FceRV8vqA%2FrZtzLEo%3D",
               oauth_token="00076e1415667a6c555f5d43582134c87d6367ab456fd2",
               oauth_timestamp="1418647040",
               realm="000068"

HTTP/1.1 200 OK
Date: Mon, 15 Dec 2014 12:37:42 GMT
Content-Type: text/html
Content-Length: 12

Hello World!
```

Using NLAuth in the Authorization Header

NLAuth passes in the following login credentials:

- `nlauth_account` (required) - NetSuite company ID
- `nlauth_email` (required) - NetSuite user name
- `nlauth_signature` (required) - NetSuite password
- `nlauth_role` (optional) - internal ID of the role used to log in to NetSuite

Note: If a user has a default role defined, this role can be used for login when the role parameter is not passed in the authorization header.

Important: NetSuite RESTlet authentication only accepts special characters that are URL encoded. If your credentials contain special characters, replace each special character with its appropriate URL encoding. For additional information on URL encoding, see http://www.w3schools.com/tags/ref_urlencode.asp

NLAuth Authorization Header Formatting

For NLAuth, the Authorization header should be formatted as:

```
NLAuth<space><comma-separated parameters>.
```

Note: NetSuite recommends entering the key value pairs without leading or trailing spaces (`nlauth_account=123456,` not `nlauth_account= 123456 ,`) as shown in the following example.

For example:

```
Authorization:  NLAuth nlauth_account=123456, nlauth_email=jsmith@ABC.com, nlauth_signature=xxx
xxxxx, nlauth_role=41
```

Important: NetSuite provides a REST roles service that you can use to determine a user's account and role. This service makes it possible to support RESTlet login when account and role are unknown, for example, in mobile applications. See Using the REST roles Service to Get User Accounts, Roles, and Domains.

RESTlet URL and Domain

The URL used for a RESTlet HTTP request varies according to whether the RESTlet is called from an external client or from a client hosted by NetSuite.

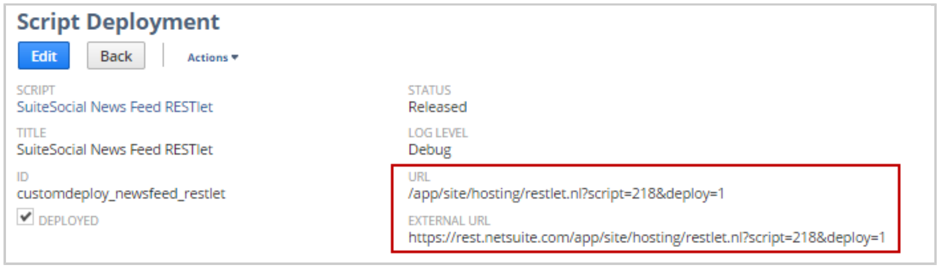
- For a RESTlet called from an external client, the URL must be an absolute URL. This URL includes the RESTlet domain specific to the data center that houses your NetSuite account data. This domain might be `https://rest.netsuite.com`, `https://rest.na1.netsuite.com`, or a similar variant. In an integration, you **must** dynamically discover this domain. If you use an incorrect domain, the request fails. To avoid this problem, use the REST roles service to dynamically discover the correct domain. For details, see Using the REST roles Service to Get User Accounts, Roles, and Domains.

- For a RESTlet called from a client hosted by NetSuite, the URL should be a relative URL. That is, the URL does not include the domain.

  When the NetSuite account is hosted at `system.netsuite.com`, the RESTlet also uses the same base URL. For example, if you use `/app/site/hosting/restlet.nl` (as seen in the following screenshot) and deploy that RESTlet in accounts located in different data centers, the application will correctly prefix `/app/site/hosting/restlet.nl` with the base URL, as appropriate for the location of each account.

The following RESTlet deployment record shows examples of URLs.

- The URL field displays the relative URL, used by NetSuite-hosted clients for any references made to objects inside NetSuite.
- The External URL field displays an absolute URL that could be used to call the RESTlet from outside of NetSuite. However, any integrations you create must dynamically discover the domain that makes up the first part of this URL (the parts that differ from the text shown under the URL field).



SCRIPT
SuiteSocial News Feed RESTlet

TITLE
SuiteSocial News Feed RESTlet

ID
customdeploy_newsfeed_restlet

✓ DEPLOYED

STATUS
Released

LOG LEVEL
Debug

URL
/app/site/hosting/restlet.nl?script=218&deploy=1

EXTERNAL URL
https://rest.netsuite.com/app/site/hosting/restlet.nl?script=218&deploy=1

Note: RESTlets use the same debugging domain as other SuiteScript types, **https://debugger.netsuite.com**. Whether the RESTlet client is hosted externally or by NetSuite does not change the debugger domain used. See RESTlet Debugging.

Using the REST roles Service to Get User Accounts, Roles, and Domains

NetSuite provides a REST roles service that returns the following data when you submit a user email address and password:

- account(s) and role(s) available to the user
- REST, web services, and general system domains to be used for external client access to NetSuite

Note: RESTlets are part of SuiteScript. They are **not** part of NetSuite's web services feature. Be aware that if a role has the Web Services Only option set to true, a user logged in through that role is permitted to send web services calls only. RESTlet calls receive an INVALID_LOGIN_CREDENTIALS error response.

The roles service fills the following needs:

- Support for RESTlet login when the user's account and role are unknown.
- Dynamic discovery of correct URLs for external client access to each NetSuite account. Discovery of the following domains is supported:

- restDomain - for example: https://rest.netsuite.com, https://rest.na1.netsuite.com, and similar variants
- systemDomain - for example: https://system.netsuite.com, https://system.na1.netsuite.com, and similar variants
- webservicesDomain - for example: https://webservices.netsuite.com, https://webservices.na1.netsuite.com, and similar variants

This discovery **is required** to support the hosting of customer accounts in multiple data centers. Each data center has a different domain, so the domain to be used for external client access depends upon the data center hosting each NetSuite account.

Important: NetSuite hosts customer accounts in multiple data centers. For that reason, the correct domain for external client RESTlet access varies depending on the data center hosting the account. Your integration should **must** incorporate logic that dynamically determines the correct RESTlet domain. The REST roles service is designed for this purpose. For details on using this service, see Using the REST roles Service to Get User Accounts, Roles, and Domains.

Additionally, if you have a web services integration that uses the 2012.1 or an earlier WSDL, you must use the roles service to discover the web services URL for your account. For sample web services code that calls the REST roles service, see the help topics Java REST Example for Web Services Domain Discovery and .NET REST Example for Web Services Domain Discovery. If you use the 2012.2 endpoint or later, you can use a web services operation, getDataCenterUrls, to obtain the appropriate web services URL.

Using the REST roles service

You can use the REST roles service to retrieve role and domain information from a production, release preview, or sandbox environment.

You call the service by sending a request to one of the roles service URLs. For examples, see the following table.

Environment	URL
Production and EU Sandbox (sandboxes hosted in European Union data centers)	https://rest.netsuite.com/rest/roles and similar variants
Release Preview	https://rest.beta.netsuite.com/rest/roles
EU Release Preview (Release Preview for accounts hosted in North American data centers)	https://rest.eu1.netsuite.com/rest/roles
NA Sandbox (sandboxes hosted in North American data centers)	https://rest.sandbox.netsuite.com/rest/roles

Your request should use the get method, and you must also include an NLAuth authorization header. This header must identify a user and can optionally identify a NetSuite account ID. In response, the service returns data about roles and domains. For information about NLAuth, see Using NLAuth in the Authorization Header.

If your request includes a NetSuite account ID, the information returned is specific to that account. Otherwise, the system returns details for each NetSuite account to which the user identified in the header has access.

Note that if you want to retrieve Customer Center roles for any user, you **must** include a NetSuite account ID in the authorization header. If you do not, the service does not return the Customer Center role, nor does it return any custom roles that were created based on the Customer Center role.

Important: If you have an integration that makes requests to the production version of the roles service, and these requests are specific to a NetSuite account, your integration must include logic for handling redirection. This logic is necessary because of how the system handles requests specific to a NetSuite account. NetSuite redirects these calls to the instance of the service specific to the data center that hosts the account. For this reason, your integration must include logic for handling the 302 Found response status code, which is the code used when redirection occurs. By contrast, if your authorization header omits a NetSuite account ID, the request is handled without redirection.

Sample REST roles Request

To get the available accounts, roles, and domains for a user, submit the user's email address and password in the authorization header, as shown in the following example:

```
URL: https://rest.netsuite.com/rest/roles

Headers:
GET /rest/roles HTTP/1.1
Accept: */*
Accept-Language: en-us
Authorization: NLAuth nlauth_email=johnsmith@xxxxx.com, nlauth_signature=****
```

Note: For information about NLAuth, see Using NLAuth in the Authorization Header.

Sample REST roles Response

The roles service returns a list of account(s) available to the user, and for each account, the associated roles and domains, as shown in the following example:

```
[{"account":{"internalId":"1234567","name":"Test Account1"},"role":{"internalId":3,"name":"Admi
nistrator"},"dataCenterURLs":{"restDomain":"https://rest.netsuite.com","systemDomain":"https://
system.netsuite.com","webservicesDomain":"https://webservices.netsuite.com"}},
{"account":{"internalId":"1234678","name":"Test Account2"},"role":{"internalId":3,"name":"Admin
istrator"},"dataCenterURLs":{"restDomain":"https://rest.netsuite.com","systemDomain":"https://s
ystem.netsuite.com","webservicesDomain":"https://webservices.netsuite.com"}},
{"account":{"internalId":"1234789","name":"Test Account3"},"role":{"internalId":3,"name":"Admin
istrator"},"dataCenterURLs":{"restDomain":"https://rest.netsuite.com","systemDomain":"https://s
ystem.netsuite.com","webservicesDomain":"https://webservices.netsuite.com"}}]
```

The role to use for login can be selected from this list. And you can use this list to determine the domains to specify in RESTlet and web services requests.

Supported Input and Output Content Types for RESTlets

RESTlets support JSON and plain text content types for input and output. For each RESTlet, output content type is the same as input content type.

You must set the content type in the HTTP Content-Type header. You can use the following values to specify the input/output content type for a RESTlet:

- application/json
- text/plain

If you specify a content type other than JSON or text, a 415 error is returned with the following message:

```
Invalid content type.  You can only use application/json or text/plain with RESTlets.
```

Using JSON Objects and Arrays

JSON is an acronym for JavaScript Object Notation, which is a subset of JavaScript. This special object notational construct is a syntax used to pass JavaScript objects containing name/value pairs, arrays, or other objects. The following JSON formatting is used for RESTlets:

- Each JSON object is an unordered set of name/value pairs, or members, enclosed in curly braces.
  - Each member is followed by a comma, which is called a value separator.
  - Within each member, the name is separated from the value by a colon, which is called a name separator.
  - Each name and each value is enclosed in quotation marks.
- Each JSON array is an ordered sequence of values, enclosed in square braces. Array values are separated by commas.

For examples of how to format JSON input for restlets, see Sample RESTlet Input Formats.

Supported Functions for RESTlets

RESTlets currently support a subset of HTTP methods, as shown in the following table:

HTTP Method	Input	Output	Description
GET	Parameter Object	Object	Requests a representation of the specified resource.
POST	Object	Object	Submits data to be processed, for example from an HTML form. Data is included in the body of the request, and can result in creation of a new resource, updates of existing resource(s), or both.
DELETE	Parameter Object	No Content	Passes in the ID and record type of the resource to be deleted, so that nlapiDeleteRecord or other delete-related logic can be called. This method does not return anything.
PUT	Object	Object	Uploads a representation of the specified resource.

The functions that implement these methods are specified on a RESTlet's script record. Each RESTlet must have a function for at least one of these HTTP methods. Each HTTP method can call any SuiteScript nlapi functions. See Create the RESTlet Script Record.

For examples of these functions in RESTlets, see Sample RESTlet Code.

RESTlet Governance and Session Management

The SuiteScript governance model tracks usage units on two levels: API level and script level. At the API level, RESTlets have the same usage limits as other types of SuiteScripts. At the script level, RESTlets allow 5,000 usage units per script, a limit five times greater than Suitelets and most other types of SuiteScripts. For more information, see the help topic SuiteScript Governance.

There is a limit of 10MB per string used as RESTlet input or output.

SuiteScript currently does not support a logout operation similar to the one used to terminate a session in SuiteTalk.

RESTlet Debugging

You can use the SuiteScript Debugger to debug RESTlet scripts, in the same manner that you use it to debug other types of server SuiteScripts. RESTlets use the same debugging domain as other SuiteScript types, **https://debugger.netsuite.com**. Both ad-hoc debugging and deployed debugging are supported for RESTlets. For general instructions for using the Debugger, see the help topic Working with the SuiteScript Debugger.

Important: For deployed debugging of a RESTlet, you need to set the session cookies of the client application that run the RESTlet to the same cookies listed for the RESTlet in the Debugger. Also, you must remove the authorization header from your RESTlet before debugging. For more details, see Debugging a RESTlet.

RESTlet Error Handling

RESTlets return standard HTTP status codes for their contained HTTP requests. A standard success code is returned for a successful request. Standard error codes are returned for errors due to unparsable input, authentication failure, lack of server response, use of an unsupported method, and use of an invalid content type or data format for input. In most cases, generic HTTP error messages are returned. The format used for error messages is the same as the specified format for input: JSON or plain text. For more details, see RESTlet Status Codes and Error Message Formats.

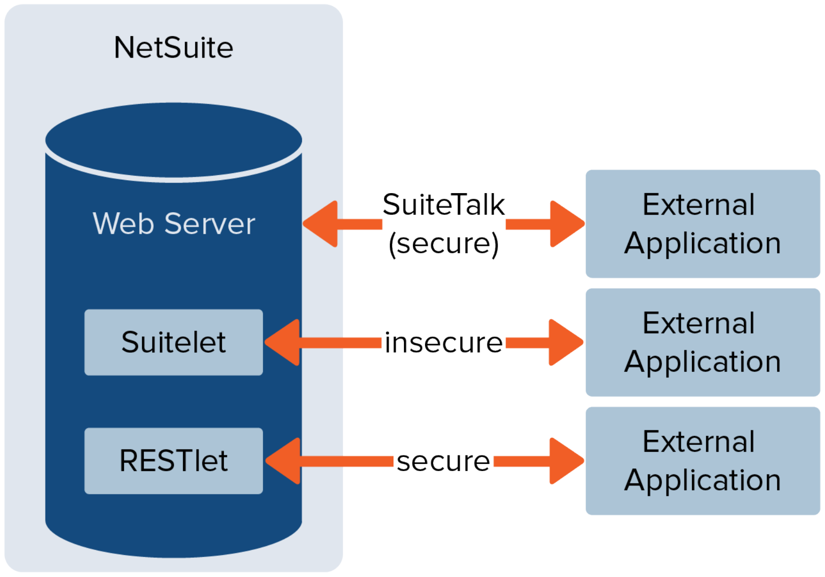
RESTlets also support the SuiteScript nlapiCreateError function. You can include this API in your code to abort script execution when an error occurs. For details, see the help topic Error Handling APIs.

RESTlet Security

The URLs for RESTlet HTTP requests can vary, but all resources are protected by TLS encryption. NetSuite supports TLS 1.0, 1.1, and 1.2 encryption for `forms.netsuite.com`, `system.netsuite.com`, and other NetSuite domains. Only requests sent using TLS encryption are granted access. For more on RESTlet domains, see RESTlet URL and Domain.

RESTlets vs. Other NetSuite Integration Options

RESTlets provide one option for integration with NetSuite. Other options include SOAP-based web services through SuiteTalk, and Suitelets.



Review the following for comparisons of these integration options:

- RESTlets Compared to SuiteTalk
- RESTlets Compared to Suitelets

RESTlets Compared to SuiteTalk

The following table compares the characteristics of RESTlets with those of SuiteTalk's SOAP-based web services:

Attribute	RESTlets	SuiteTalk
Supported Operations	get, search, add, update (heterogeneous)	get, search, add, update (homogenous)
Authentication Supported?	Yes	Yes
Supported HTTP Methods	GET, PUT, POST, DELETE	POST
Passing of Login Details	in authorization header	in body (SOAP)
Passing of Parameters	GET parameters on URL	all parameters in body (SOAP)
Supported Content Types	JSON, text/xml (explicit)	text/xml (explicit)

Attribute	RESTlets	SuiteTalk
Environment	lightweight, more suitable for mobile devices, bundleable	heavy programming and deployment environment (C#, Java)
URL Clarity?	Yes https://rest.netsuite.com/app/site/hosting/restlet.nl?script=57&deploy=1&recordtype=salesorder&id=21480 (Note that for clients hosted by NetSuite, use the relative URL that does not include the domain.)	No https://webservices.netsuite.com/services/NetSuitePort_2011_1

Note: SuiteTalk is recommended for system-to-system integrations.

RESTlets Compared to Suitelets

The following table compares the characteristics of RESTlets with those of Suitelets:

Attribute	RESTlets	Suitelets
Supported Operations	get, search, add, update	get, search, add, update
Authentication Supported?	Yes	No , when available without login and executed as admin programmatically Yes, when accessed from a browser by a logged-in NetSuite user
Script Functions and HTTP Methods	individual script function for each HTTP method	one script function for all HTTP method
Content Handling	built-in handling of JSON input/output	must write code to convert JSON input/output
Governance	5,000 usage units per script	1,000 usage units per script
URL Clarity?	Yes https://rest.netsuite.com/app/site/hosting/restlet.nl?script=57&deploy=1&recordtype=salesorder&id=21480 (Note that for clients hosted by NetSuite, use the relative URL that does not include the domain.)	No https://forms/netsuite.com/app/site/hosting/scriptlet.nl?script=62&deploy=1&compid=824056&h=ec041b59b3075bec783d

Creating a RESTlet

To run a RESTlet in NetSuite, you must first define your client code and behavior, then define your RESTlet and its required functions. The client will send requests to the RESTlet URL generated by NetSuite.

To define a RESTlet:

1. Create a JavaScript file for your RESTlet code.
2. Load the file into NetSuite.
3. Create a script record where you define SuiteScript functions for one or more HTTP methods.
4. Define all runtime options on the Script Deployment page.

See the following for instructions for these tasks:

- Create the RESTlet File and Add It to the File Cabinet
- Create the RESTlet Script Record
- Define RESTlet Deployment(s)

Create the RESTlet File and Add It to the File Cabinet

1. Create a .js file and add your code to it, in the same manner that you create other types of SuiteScript files, as described in Step 1: Create Your Script.

   This single script file should include GET, POST, DELETE, or PUT function(s) as necessary.
2. After you have created a .js file with your RESTlet code, you need to add this file to the NetSuite file cabinet.

   The following steps describe how to add the file manually. If you are using SuiteCloud IDE, this process is automated. For more information, see the help topic Step 2: Add Script to NetSuite File Cabinet.

   1. Go to Documents > Files > File Cabinet, and select the folder where you want to add the file.

      It is recommended that you add your file to the SuiteScripts folder, but it can be added to any other folder of your choice.
   2. Click **Add File**, and browse to the .js file.

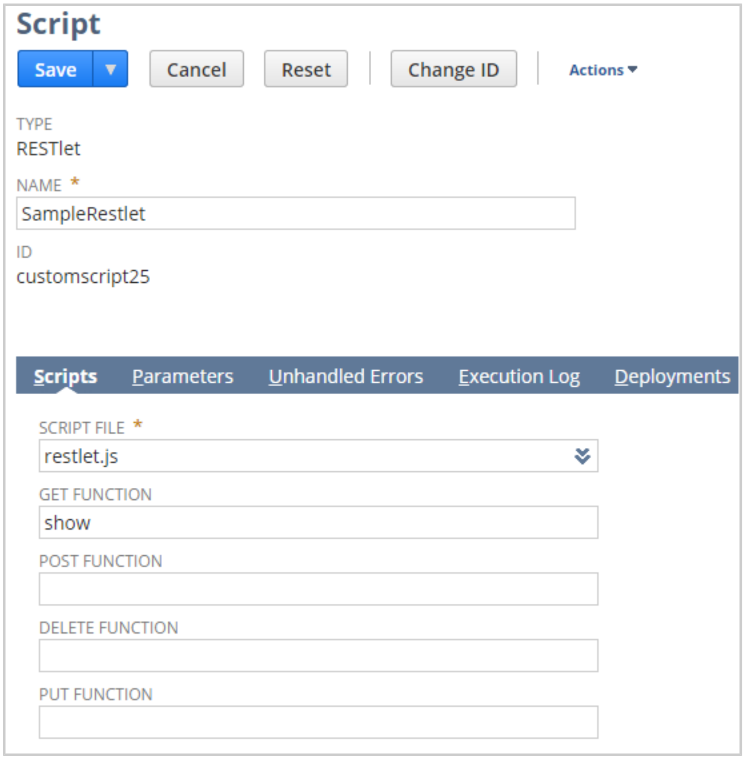
Create the RESTlet Script Record

After you have added a RESTlet file to the file cabinet, you can create a NetSuite script record.

To create a RESTlet script record:

1. Go to Setup > Customization > Scripts > New, and click **RESTlet**.
2. Complete fields in the script record and save.

   Although you do not need to set every field on the Script record, at a minimum you must provide a **Name** for the Script record, load your SuiteScript file to the record, and specify at least one of the following executing functions on the Scripts tab: GET, POST, DELETE, or PUT.



TYPE
RESTlet

NAME *
SampleRestlet

ID
customscript25

SCRIPT FILE *
restlet.js

GET FUNCTION
show

POST FUNCTION

DELETE FUNCTION

PUT FUNCTION

You can specify more than one of these functions as desired. These functions should all be in the main script file. If these functions call functions in other script files, you need to list those files as library script files.

For more details about creating a script record, see the help topic Steps for Creating a Script Record.

Define RESTlet Deployment(s)

After you have created a RESTlet script record, you need to define at least one deployment. For details about defining script deployments, see the help topics Step 5: Define Script Deployment and Steps for Defining a Script Deployment

You can define multiple deployments per RESTlet.

To define a RESTlet script deployment.

1. Do one of the following:
   - When you save your Script record, you can immediately create a Script Deployment record by selecting **Save and Deploy** from the Script record **Save** button.
   - If you clicked **Save**, immediately afterwards you can click **Deploy Script** on the script record.
   - If you want to update a deployment that already exists, go to Customization > Scripting > Script Deployments > [deployment] > Edit.
2. Complete fields in the script deployment record and click **Save**.

   If you want to debug the script, set the **Status** to **Testing**.

Note: After you have saved a RESTlet deployment, the deployment record includes the URL used to invoke the RESTlet. For a RESTlet called from an externally hosted client, use the **External URL**. For a RESTlet called from a client hosted by NetSuite, use the **URL** that does not include the domain. See RESTlet URL and Domain.

Debugging a RESTlet

You can use the NetSuite Debugger to debug RESTlet code in the same manner that you debug other types of SuiteScript code, as described in the NetSuite Help Center topic Debugging SuiteScript.

- To debug code snippets before you have a RESTlet script record that has been deployed, called ad-hoc debugging, follow the instructions in Ad Hoc Debugging . (Be sure not to include the RESTlet's authorization header in the code snippets to be debugged, as this header can prevent the debugger from working.)
- To debug an existing script that has a defined deployment, called deployed debugging, follow the steps below.

Important: In addition to debugging RESTlet script code, it is recommended that you test the HTTP request to be sent to the RESTlet. Free tools are available for this purpose. See RESTlet HTTP Testing Tools.

To debug a deployed RESTlet:

1. Before you deploy a RESTlet to be debugged, ensure that the script does not include the HTTP authorization header, as this header can prevent the debugger from working.
2. Ensure that on the script deployment record, the **Status** value is set to **Testing**.
3. Go to Customization > Scripting > Script Debugger, or log in to the debugger domain **https://debugger.netsuite.com**. (See the help topic Deployed Debugging for details.)
4. Click the **Debug Existing** button in the main Script Debugger page.

   Note: This button only appears when you have deployed scripts with the status is set to **Testing**.
5. Select the RESTlet script that you want to debug in the Script Debugger popup.

   After you click the **Select** option button, the RESTlet's cookies display in a banner.
6. Copy the cookies and paste them into a text file so that you have them available.
7. Click the **Select and Close** button in the Script Debugger popup.

   The main Script Debugger page displays a message that it is waiting for user action.
8. Set the cookies in your client application to the values you copied in step 5, and send the RESTlet request.

   The main Script Debugger page displays the script execution as pending at the NetSuite function `restletwrapper(request)`.
9. You have the following options for debugging your script code:
   - Click the **Step Over** button to begin stepping through each line of code.
   - Add watches and evaluate expressions.
   - Set break points and click the **Run** button to run the code. The Debugger will stop code execution at the first break point set.
   - Set no break points, click the **Run** button, and have the Debugger execute the entire piece of code.

See the help topic SuiteScript Debugger Interface for information on stepping into/out of functions, adding watches, setting and removing break points, and evaluating expressions.

Debugging Timeout Errors

If a timeout error occurs during debugging, check for the following:

- Invalid or missing NetSuite version

  Ensure that you have correctly copied the session cookies, as described in the steps above. These cookies includes a valid NetSuite version.
- Incorrect domain such as https://rest.netsuite.com

  Ensure that you have logged in to https://debugger.netsuite.com.

RESTlet HTTP Testing Tools

You can use the tools of your choosing to test the HTTP request to be sent to a RESTlet.

If you have installed and set up SuiteCloud IDE, a debug client is available for your use. The RESTlet/Suitelet Debug Client enables you to debug deployed RESTlet and Suitelet SuiteScripts with the SuiteCloud IDE Debugger. The client is only accessible after a debug session is started. For details about this client, see the help topic Using the RESTlet/Suitelet Debug Client. For information about SuiteCloud IDE, see the help topic Working with SuiteCloud IDE.

In addition, the following free tools are available:

- **Send HTTP Tool**

  This tool is a free HTTP Request builder that you can use to send an HTTP request to the RESTlet and analyze the response.

  http://soft-net.net/SendHTTPTool.aspx
- **Fiddler**

  This tool is a free Web debugging proxy that you can use to log HTTP traffic, and inspect the HTTP request and response for the RESTlet.

  http://www.fiddler2.com/fiddler2/

Warning: The above information about free tools is provided as a courtesy and is not intended as an endorsement or recommendation of these tools.

Sample RESTlet Code

The following examples provide sample RESTlet code:

- Simple Example to Get Started
- Example Code Snippets of HTTP Methods
- Example RESTlet Called from a Portlet Script
- Example RESTlet Request from Android
- Example RESTlet Request Using nlapiRequestURL

Simple Example to Get Started

Use the following example as a very basic GET method test when you are getting started with RESTlets:

```
function sayhi()
{
    var o = new Object();
    o.sayhi = 'Hello World! ';
    return o;
}
```

Example Code Snippets of HTTP Methods

The following code snippets provide examples of RESTlet functions.

GET Method

```
// Get a standard NetSuite record
function getRecord(datain)
{
    return nlapiLoadRecord(datain.recordtype, datain.id); // e.g recordtype="customer", id=769
}
```

Query parameters:

```
 recordtype=customer&id=769
```

POST Method

```
// Create a standard NetSuite record
function createRecord(datain)
{
    var err = new Object();

    // Validate if mandatory record type is set in the request
    if (!datain.recordtype)
    {
        err.status = "failed";
        err.message= "missing recordtype";
        return err;
    }

    var record = nlapiCreateRecord(datain.recordtype);

    for (var fieldname in datain)
    {
     if (datain.hasOwnProperty(fieldname))
     {
         if (fieldname != 'recordtype' && fieldname != 'id')
         {
             var value = datain[fieldname];
             if (value && typeof value != 'object') // ignore other type of parameters
             {
                 record.setFieldValue(fieldname, value);
             }
         }
         }
    }
    var recordId = nlapiSubmitRecord(record);
    nlapiLogExecution('DEBUG','id='+recordId);

    var nlobj = nlapiLoadRecord(datain.recordtype,recordId);
    return nlobj;
}
```

Request Payload:

```
{"recordtype":"customer","entityid":"John Doe","companyname":"ABCTools Inc","subsidiary":"1","e
mail":"jdoe@email.com"}
```

DELETE Method

```
// Delete a standard NetSuite record
function deleteRecord(datain)
{
    nlapiDeleteRecord(datain.recordtype, datain.id); // e.g recordtype="customer", id="769"
}
```

Query parameters:

```
recordtype=customer&id=769
```

Example RESTlet Called from a Portlet Script

- Portlet Script Code - Calls RESTlet and Sets Search Criteria
- RESTlet Code - Gets Data based on Portlet Script Criteria

Portlet Script Code - Calls RESTlet and Sets Search Criteria

```
function getAccount () { return '1234567'; }

function getRESTletURL()
{
   return 'https://rest.netsuite.com/app/site/hosting/restlet.nl?script=27&deploy=1&c='+getAcco
unt(); // for phasing
}

function credentials()
{
    this.email='jsmith@abcauto.com';
    this.account=getAccount();
    this.role='3';
    this.password='mysecretpwd';
}

// Portlet function
function displayOpportunities(portlet)
{
   portlet.setTitle('Opportunities');
        var col = portlet.addColumn('id','text', 'ID', 'LEFT');
        col.setURL(nlapiResolveURL('RECORD','opportunity'));
        col.addParamToURL('id','id', true);
        portlet.addColumn('title','text', 'Opportunity', 'LEFT');
        portlet.addColumn('customer','text', 'Customer', 'LEFT');
        portlet.addColumn('salesrep','text', 'Sales Rep', 'LEFT');
        portlet.addColumn('amount','currency', 'Amount', 'RIGHT');
        portlet.addColumn('probability','currency', 'Probability', 'RIGHT');

   var opps = getOpportunites();
   if ( opps != null && opps.length > 0 )
   {
      for ( var i=0; i < opps.length ; i++ )
      {
          portlet.addRow(opps[i]);
      }
   }
}

function getOpportunites()
{
   var url = getRESTletURL() + '&daterange=daysAgo90&&probability=10';
   var cred = new credentials();

   var headers = new Array();
   headers['User-Agent-x'] = 'SuiteScript-Call';
   headers['Authorization'] = 'NLAuth nlauth_account='+cred.account+', nlauth_email='+cred.emai
l+', nlauth_signature='+cred.password+', nlauth_role='+cred.role;
   headers['Content-Type'] = 'application/json';

   var response = nlapiRequestURL( url, null, headers );

        var responsebody = JSON.parse(response.getBody());

   var error = responsebody['error'];
        if (error)
        {
             var code = error.code;
             var message = error.message;
             nlapiLogExecution('DEBUG','failed: code='+code+'; message='+message);
             nlapiCreateError(code, message, false);
        }

        return responsebody['nssearchresult'];
}

function opportunity(internalid, title, probability, amount, customer, salesrep)
{
   this.id = internalid;
   this.title = title;
   this.probability = probability;
   this.amount = amount;
   this.customer = customer;
   this.salesrep = salesrep;
```

```
}
```

RESTlet Code - Gets Data based on Portlet Script Criteria

```
function opportunity(internalid, title, probability, amount, customer, salesrep)
{
   this.id = internalid;
   this.title = title;
   this.probability = probability;
   this.amount = amount;
   this.customer = customer;
   this.salesrep = salesrep;
}

// RESTlet Get NetSuite record data
function getRecords(datain)
{
    var filters = new Array();
    var daterange = 'daysAgo90';
    var projectedamount = 0;
    var probability = 0;
    if (datain.daterange) {
        daterange = datain.daterange;
    }
    if (datain.projectedamount) {
        projectedamount = datain.projectedamount;
    }
    if (datain.probability) {
        probability = datain.probability;
    }

    filters[0] = new nlobjSearchFilter( 'trandate', null, 'onOrAfter', daterange ); // like day
sAgo90
    filters[1] = new nlobjSearchFilter( 'projectedamount', null, 'greaterthanorequalto', projec
tedamount);
    filters[2] = new nlobjSearchFilter( 'probability', null, 'greaterthanorequalto', probabilit
y );

    // Define search columns
    var columns = new Array();
    columns[0] = new nlobjSearchColumn( 'salesrep' );
    columns[1] = new nlobjSearchColumn( 'expectedclosedate' );
    columns[2] = new nlobjSearchColumn( 'entity' );
    columns[3] = new nlobjSearchColumn( 'projectedamount' );
    columns[4] = new nlobjSearchColumn( 'probability' );
    columns[5] = new nlobjSearchColumn( 'email', 'customer' );
    columns[6] = new nlobjSearchColumn( 'email', 'salesrep' );
    columns[7] = new nlobjSearchColumn( 'title' );

    // Execute the search and return results

    var opps = new Array();
    var searchresults = nlapiSearchRecord( 'opportunity', null, filters, columns );
    // Loop through all search results. When the results are returned, use methods
    // on the nlobjSearchResult object to get values for specific fields.
    for ( var i = 0; searchresults != null && i < searchresults.length; i++ )
    {
        var searchresult = searchresults[ i ];
        var record = searchresult.getId( );
        var salesrep = searchresult.getValue( 'salesrep' );
        var salesrep_display = searchresult.getText( 'salesrep' );
        var salesrep_email = searchresult.getValue( 'email', 'salesrep' );
        var customer = searchresult.getValue( 'entity' );
        var customer_display = searchresult.getText( 'entity' );
        var customer_email = searchresult.getValue( 'email', 'customer' );
        var expectedclose = searchresult.getValue( 'expectedclosedate' );
        var projectedamount = searchresult.getValue( 'projectedamount' );
        var probability = searchresult.getValue( 'probability' );
        var title = searchresult.getValue( 'title' );

        opps[opps.length++] = new opportunity( record,
               title,
               probability,
               projectedamount,
               customer_display,
               salesrep_display);
    }

    var returnme = new Object();
    returnme.nssearchresult = opps;
    return returnme;
}
```

Example RESTlet Request from Android

```
HttpPost post = new HttpPost( URL + urlParams );

HttpParams httpParameters = new BasicHttpParams();
HttpConnectionParams.setConnectionTimeout( httpParameters, 20000 );
HttpConnectionParams.setSoTimeout( httpParameters, 42000 );

String authorization = "NLAuth nlauth_account=" + account + ", nlauth_email=" + email + ", nlau
th_signature="+password+", nlauth_role="+role+"";
post.setHeader( "Authorization", authorization );
post.setHeader( "Content-Type", "application/json" );
post.setHeader( "Accept", "*/*" );

post.setEntity( new StringEntity( "{\"name\":\"John\"}" /*input data*/ ) );

HttpClient client = new DefaultHttpClient( httpParameters );
BufferedReader in = null;

HttpResponse response = client.execute( post );

in = new BufferedReader( new InputStreamReader( response.getEntity().getContent() ) );
StringBuffer sb = new StringBuffer( "" );
```

```
String line;
String NL = System.getProperty( "line.separator" );
while ( (line = in.readLine()) != null )
{
                sb.append( line + NL );
}
in.close();
String result = sb.toString();
```

Example RESTlet Request Using nlapiRequestURL

```
function credentials(){
    this.email = "msmith@email.com";
    this.account = "1234567";
    this.role = "3";
    this.password = "*****";
}

function replacer(key, value){
    if (typeof value == "number" && !isFinite(value)){
        return String(value);
    }
    return value;
}

//Setting up URL
var url = "https://rest.netsuite.com/app/site/hosting/restlet.nl?script=260&deploy=1";

//Calling credential function
var cred = new credentials();

//Setting up Headers
var headers = {"User-Agent-x": "SuiteScript-Call",
               "Authorization": "NLAuth nlauth_account=" + cred.account + ", nlauth_email=" + c
red.email +
                                 ", nlauth_signature= " + cred.password + ", nlauth_role=" + cre
d.role,
               "Content-Type": "application/json"};

//Setting up Datainput
var jsonobj = {"recordtype": "customer",
               "entityid": "John Doe",
               "companyname": "ABC Company",
               "subsidiary": "1",
               "email": "jdoe@email.com"}

//Stringifying JSON
var myJSONText = JSON.stringify(jsonobj, replacer);

var response = nlapiRequestURL(url, myJSONText, headers);

//Below is being used to put a breakpoint in the debugger
var i=0;
```

```
//**************RESTLET Code****************

// Create a standard NetSuite record
function createRecord(datain)
{
    var err = new Object();

    // Validate if mandatory record type is set in the request
    if (!datain.recordtype)
    {
        err.status = "failed";
        err.message = "missing recordtype";
        return err;
    }

    var record = nlapiCreateRecord(datain.recordtype);

    for (var fieldname in datain)
    {
        if (datain.hasOwnProperty(fieldname))
        {
            if (fieldname != 'recordtype' && fieldname != 'id')
            {
                var value = datain[fieldname];
                if (value && typeof value != 'object') // ignore other type of parameters
                {
                    record.setFieldValue(fieldname, value);
                }
            }
        }
    }
    var recordId = nlapiSubmitRecord(record);
    nlapiLogExecution('DEBUG','id='+recordId);

    var nlobj = nlapiLoadRecord(datain.recordtype,recordId);
    return nlobj;
}
```

Sample RESTlet Input Formats

The following examples illustrate how to format input for RESTlets for the JSON content type:

- Customer Record Format
- Item Record Format
- Item Pricing Formats
- Sales Order Record Format

For a general explanation of JSON, see Using JSON Objects and Arrays.

Customer Record Format

JSON

```
{
    "shipcomplete":false,
    "giveaccess":false,
    "globalsubscriptionstatus":"1",
    "isperson":false,
    ... more body fields...,
    "consoldepositbalance":0.00,
    "entityid":"John Doe",
    "addressbook":
    [
        {"zip":"94404","phone":"650-627-1000"},
        {"zip":"94403","phone":"650-627-1001"}
    ],
    "consoloverduebalance":0.00,
    "overduebalance":0.00,
    "creditholdoverride":"AUTO",
    "resubscribelink":"Send Subscription Email"
}
```

XML

```
<shipcomplete>F</shipcomplete>
<globalsubscriptionstatus>1</globalsubscriptionstatus>
<giveaccess>F</giveaccess>
<isperson>F</isperson>
<datecreated>12/19/2010 10:26 pm</datecreated>
<salesrep>-5</salesrep><currency>1</currency>
<lastmodifieddate>12/20/2010 10:14 am</lastmodifieddate>
<id>1185</id>
<emailtransactions>F</emailtransactions>
<balance>0.00</balance>
<entitystatus>13</entitystatus>
<isbudgetapproved>F</isbudgetapproved>

... more fields...

<entityid>John Doe</entityid>
<isinactive>F</isinactive>
<addressbook>
   <zip>94404</zip>
   <phone>650-627-1000</phone>
   <defaultshipping>T</defaultshipping>
   <addrtext>Netsuite100 Mission StreetFoster City CA 94404United States</addrtext>
   <state>CA</state>
   <addressee>Netsuite</addressee>
   <isresidential>F</isresidential>
   <label>Home</label>
   <city>Foster City</city>
   <country>US</country>
   <displaystate>California</displaystate>
   <dropdownstate>CA</dropdownstate>
   <addr1>100 Mission Street</addr1>
   <override>F</override>
   <defaultbilling>T</defaultbilling>
</addressbook>
<addressbook>
   <zip>94403</zip>
   <phone>650-627-1001</phone>
   <defaultshipping>F</defaultshipping>
   <addrtext>Netsuite2955 Campus DriveSan Mateo CA 94403United States</addrtext>
   <state>CA</state>
   <addressee>Netsuite</addressee>
   <isresidential>F</isresidential>
   <label>Work</label>
   <city>San Mateo</city>
   <country>US</country>
   <displaystate>California</displaystate>
   <dropdownstate>CA</dropdownstate>
   <addr1>2955 Campus Drive</addr1>
   <override>F</override>
   <defaultbilling>F</defaultbilling>
</addressbook>
```

Item Record Format

Note: The format for item pricing varies according to the related features that are enabled in your account. See Item Pricing Formats for examples.

JSON

```
{
    "salesdescription":"Cat 5 Patch Cable 10 ft",
    "vendorname":"CABL0002-64",
    "averagecost":3.50,

    ... more fields...,

“pricing”:
[

{
                        "currency":
                        {
                                    "name":"British pound",
                                    "internalid":"2"
                        },
                        “pricelist”:
                        [
                                    {
                                    "pricelevel":
                                    {
                                                "name":"Alternate Price 1",
                                                "internalid":"2"
                                    },
                                    "price":
                                    [
                                                {
                                                            "price":9.03,
                                                            "quantitylevel":"1",
                                                            "quantity":0
                                                },
                                                {
                                                            "price":8.55,
                                                            "quantitylevel":"2",
                                                            "quantity":10
                                                }
                                    ],
                                    "discount":
                                    {
                                                "name":"-5.0%",
                                                "value":"-5.0%"
                                    }
                        },
                        {
                                    "pricelevel":
                                    {
                                                "name":"Alternate Price 2",
                                                "internalid":"3"
                                    },
                                    "price":
                                    [
                                                {
                                                            "price":8.55,
                                                            "quantitylevel":"1",
                                                            "quantity":0
                                                },
                                                {
                                                            "price":8.10,
                                                            "quantitylevel":"2",
                                                            "quantity":10
                                                }
                                    ],
                                    "discount":
                                    {
                                                "name":"-10.0%",
                                                "value":"-10.0%"
                                    }
                        },

        ]
}
Repeat for other currencies
],
    "productfeed":["FROOGLE","SHOPPING","SHOPZILLA","NEXTAG","YAHOO"],
    "weight":"1",
    "itemid":"Cable - Cat 5, 10 ft",

    ... more fields...,

    "availabletopartners":false,
    "sitecategory":
    [
   {"categorydescription":"Cables",
   "category":"12",
   "isdefault":false}
    ],
    "costingmethoddisplay":"Average",
    "offersupport":true
}
```

XML

```
<salesdescription>Cat 5 Patch Cable 10 ft</salesdescription>
<vendorname>CABL0002-64</vendorname>

... more ...

<excludefromsitemap>F</excludefromsitemap>
<isdonationitem>F</isdonationitem>
<recordtype>inventoryitem</recordtype>
<createddate>10/12/2006 8:37 pm</createddate>
<cost>3.50</cost>
<price4>
   <pricelevelname>Base Price</pricelevelname>
</price4>
<price4>
   <pricelevelname>Alternate Price 3</pricelevelname>
</price4>
<price4>
   <discountdisplay>-10.0%</discountdisplay>
   <pricelevelname>Corporate Discount Price</pricelevelname>
</price4>
<price4>
   <discountdisplay>-15.0%</discountdisplay>
   <pricelevelname>Employee Price</pricelevelname>
</price4>
<price4>
   <pricelevelname>Online Price</pricelevelname>
</price4>
<price3>
   <pricelevelname>Base Price</pricelevelname>
</price3>
<price3>
   <pricelevelname>Alternate Price 3</pricelevelname>
</price3>
<price3>
   <discountdisplay>-10.0%</discountdisplay>
   <pricelevelname>Corporate Discount Price</pricelevelname>
</price3>
<price3>
   <discountdisplay>-15.0%</discountdisplay>
   <pricelevelname>Employee Price</pricelevelname>
</price3>
<price3>
   <pricelevelname>Online Price</pricelevelname>
</price3>
<price2>
   <price[1]>5.00</price[1]>
   <pricelevelname>Base Price</pricelevelname>
   <price[2]>4.00</price[2]>
</price2>
<price2>
   <pricelevelname>Alternate Price 3</pricelevelname>
</price2>
<price2>
   <discountdisplay>-10.0%</discountdisplay>
   <price[1]>4.50</price[1]>
   <pricelevelname>Corporate Discount Price</pricelevelname>
   <price[2]>3.60</price[2]>
</price2>
<price2>
   <discountdisplay>-15.0%</discountdisplay>
   <price[1]>4.25</price[1]>
   <pricelevelname>Employee Price</pricelevelname>
   <price[2]>3.40</price[2]>
</price2>
<price2>
   <pricelevelname>Online Price</pricelevelname>
</price2>
<price1>
   <price[1]>10.95</price[1]>
   <pricelevelname>Base Price</pricelevelname>
   <price[2]>10.00</price[2]>
</price1>
<price1>
   <pricelevelname>Alternate Price 3</pricelevelname>
</price1>
<price1>
   <discountdisplay>-10.0%</discountdisplay>
   <price[1]>9.86</price[1]>
   <pricelevelname>Corporate Discount Price</pricelevelname>
   <price[2]>9.00</price[2]>
</price1>
<price1>
   <discountdisplay>-15.0%</discountdisplay>
   <price[1]>9.31</price[1]>
   <pricelevelname>Employee Price</pricelevelname>
   <price[2]>8.50</price[2]>
</price1>
<price1>
   <price[1]>10.95</price[1]>
   <pricelevelname>Online Price</pricelevelname>
   <price[2]>10.00</price[2]>
</price1>
<productfeed>FROOGLE</productfeed>
<productfeed>SHOPPING</productfeed>
<productfeed>SHOPZILLA</productfeed>
<productfeed>NEXTAG</productfeed>
<productfeed>YAHOO</productfeed>
<weight>1</weight>
<itemid>Cable - Cat 5, 10 ft</itemid>

... more fields...

<sitecategory>
    <category>12</category>
    <categorydescription>Cables</categorydescription>

    <isdefault>F</isdefault>
</sitecategory>
<offersupport>T</offersupport>
```

Item Pricing Formats

The format for item pricing varies according to which of the following features are enabled in your account: Multiple Prices, Quantity Pricing, and Multiple Currencies. The following examples show the JSON format for item pricing when these features are enabled.

- Single Price (no additional pricing features enabled)
- Multiple Prices Only Enabled
- Quantity Pricing Only Enabled
- Multiple Prices, Multiple Currencies Enabled
- Multiple Prices, Quantity Pricing, Multiple Currencies Enabled

Single Price (no additional pricing features enabled)

```
"pricing":
[
    {
        "pricelist":
        [
            {
                "price":
                [
                    {"price":100.00,"quantitylevel":"1","quantity":0}
                ]
            }
        ],
        "currency":{"name":"USA","internalid":"1"}
  }
]
```

Multiple Prices Only Enabled

```
"pricing":
[
    {
        "pricelist":
        [
            {
                "pricelevel":{"name":"Base Price","internalid":"1"},
                "price":
                [
                    {"price":100.00,"quantitylevel":"1","quantity":0}
                ]
            },
            {
                "pricelevel":{"name":"Alternate Price 1","internalid":"2"},
                "price":
                [
                    {"price":99.00,"quantitylevel":"1","quantity":0}
                ]
            },
            {
                "pricelevel":{"name":"Alternate Price 2","internalid":"3"},
                "price":
                [
                    {"price":98.00,"quantitylevel":"1","quantity":0}
                ]
            },
            {
                "pricelevel":{"name":"Alternate Price 3","internalid":"4"},
                "price":
                [
                    {"price":97.00,"quantitylevel":"1","quantity":0}
                ]
            },
            {
                "pricelevel":{"name":"Online Price","internalid":"5"},
                "price":
                [
                    {"price":96.00,"quantitylevel":"1","quantity":0}
                ]
            }
        ],
        "currency":{"name":"USA","internalid":"1"}
    }
]
```

Quantity Pricing Only Enabled

```
"pricing":
[
    {
        "pricelist":
```

```
        [
            {
                "pricelevel":{"name":"Base Price","internalid":"1"},
                "price":
                [
                    {"price":100.00,"quantitylevel":"1","quantity":0},
                    {"price":95.00,"quantitylevel":"2","quantity":100},
                    {"price":90.00,"quantitylevel":"3","quantity":150},
                    {"price":85.00,"quantitylevel":"4","quantity":200},
                    {"price":80.00,"quantitylevel":"5","quantity":250}
                ]
            },
            {
                "pricelevel":{"name":"Alternate Price 1","internalid":"2"},
                "price":
                [
                    {"price":99.00,"quantitylevel":"1","quantity":0},
                    {"price":94.00,"quantitylevel":"2","quantity":100},
                    {"price":89.00,"quantitylevel":"3","quantity":150},
                    {"price":84.00,"quantitylevel":"4","quantity":200},
                    {"price":79.00,"quantitylevel":"5","quantity":250}
                ]
            },
            {
                "pricelevel":{"name":"Alternate Price 2","internalid":"3"},
                "price":
                [
                    {"price":98.00,"quantitylevel":"1","quantity":0},
                    {"price":93.00,"quantitylevel":"2","quantity":100},
                    {"price":88.00,"quantitylevel":"3","quantity":150},
                    {"price":83.00,"quantitylevel":"4","quantity":200},
                    {"price":78.00,"quantitylevel":"5","quantity":250}
                ]
             },
             {
                 "pricelevel":{"name":"Alternate Price 3","internalid":"4"},
                 "price":
                 [
                     {"price":97.00,"quantitylevel":"1","quantity":0},
                     {"price":92.00,"quantitylevel":"2","quantity":100},
                     {"price":87.00,"quantitylevel":"3","quantity":150},
                     {"price":82.00,"quantitylevel":"4","quantity":200},
                     {"price":77.00,"quantitylevel":"5","quantity":250}
                 ]
             },
             {
                 "pricelevel":{"name":"Online Price","internalid":"5"},
                 "price":
                 [
                     {"price":96.00,"quantitylevel":"1","quantity":0},
                     {"price":91.00,"quantitylevel":"2","quantity":100},
                     {"price":86.00,"quantitylevel":"3","quantity":150},
                     {"price":81.00,"quantitylevel":"4","quantity":200},
                     {"price":76.00,"quantitylevel":"5","quantity":250}
                 ]
             }
         ],
         "currency":{"name":"USA","internalid":"1"}
     }
]
```

Multiple Prices, Multiple Currencies Enabled

```
"pricing":
[
    {
        "pricelist":
        [
            {
                "pricelevel":{"name":"Base Price","internalid":"1"},
                "price":[{"price":110.00,"quantitylevel":"1","quantity":0}]
            },
            {
                "pricelevel":{"name":"Alternate Price 1","internalid":"2"},
                "price":[{"price":105.00,"quantitylevel":"1","quantity":0}]
            },
            {
                "pricelevel":{"name":"Alternate Price 2","internalid":"3"},
                "price":[{"price":100.00,"quantitylevel":"1","quantity":0}]
            },
            {
                "pricelevel":{"name":"Alternate Price 3","internalid":"4"},
                "price":[{"price":95.00,"quantitylevel":"1","quantity":0}]
            },
            {
                "pricelevel":{"name":"Online Price","internalid":"5"},
                "price":[{"price":90.00,"quantitylevel":"1","quantity":0}]
            }
        ],
        "currency":{"name":"British pound","internalid":"2"}
    },
    {
        "pricelist":
        [
            {"pricelevel":{"name":"Base Price","internalid":"1"},"price":[{"price":100.00,"quan
titylevel":"1","quantity":0}]},
            {"pricelevel":{"name":"Alternate Price 1","internalid":"2"},"price":[{"price":99.00
,"quantitylevel":"1","quantity":0}]},
            {"pricelevel":{"name":"Alternate Price 2","internalid":"3"},"price":[{"price":98.00
,"quantitylevel":"1","quantity":0}]},
            {"pricelevel":{"name":"Alternate Price 3","internalid":"4"},"price":[{"price":97.00
,"quantitylevel":"1","quantity":0}]},
            {"pricelevel":{"name":"Online Price","internalid":"5"},"price":[{"price":96.00,"qua
ntitylevel":"1","quantity":0}]}
        ],
        "currency":{"name":"USA","internalid":"1"}
}
```

Multiple Prices, Quantity Pricing, Multiple Currencies Enabled

```
"pricing":
[
    {
        "pricelist":
        [
            {
                "pricelevel":{"name":"Base Price","internalid":"1"},
                "price":
                [
                    {"price":110.00,"quantitylevel":"1","quantity":0},
                    {"price":105.00,"quantitylevel":"2","quantity":100},
                    {"price":100.00,"quantitylevel":"3","quantity":150},
                    {"price":95.00,"quantitylevel":"4","quantity":200},
                    {"price":90.00,"quantitylevel":"5","quantity":250}
                ]
            },
            {
                "pricelevel":{"name":"Alternate Price 1","internalid":"2"},
                "price":
                [
                    {"price":105.00,"quantitylevel":"1","quantity":0},
                    {"price":100.00,"quantitylevel":"2","quantity":100},
                    {"price":95.00,"quantitylevel":"3","quantity":150},
                    {"price":90.00,"quantitylevel":"4","quantity":200},
                    {"price":85.00,"quantitylevel":"5","quantity":250}
                ]
            },
            {
                "pricelevel":{"name":"Alternate Price 2","internalid":"3"},
                "price":[{"price":100.00,"quantitylevel":"1","quantity":0},{"price":95.00,"quan
titylevel":"2","quantity":100},{"price":90.00,"quantitylevel":"3","quantity":150},{"price":85.0
0,"quantitylevel":"4","quantity":200},{"price":80.00,"quantitylevel":"5","quantity":250}]
            },
            {
                "pricelevel":{"name":"Alternate Price 3","internalid":"4"},
                "price":[{"price":95.00,"quantitylevel":"1","quantity":0},{"price":90.00,"quant
itylevel":"2","quantity":100},{"price":85.00,"quantitylevel":"3","quantity":150},{"price":80.00
,"quantitylevel":"4","quantity":200},{"price":75.00,"quantitylevel":"5","quantity":250}]
            },
            {
                "pricelevel":{"name":"Online Price","internalid":"5"},
                "price":[{"price":90.00,"quantitylevel":"1","quantity":0},{"price":85.00,"quant
itylevel":"2","quantity":100},{"price":80.00,"quantitylevel":"3","quantity":150},{"price":75.00
,"quantitylevel":"4","quantity":200},{"price":70.00,"quantitylevel":"5","quantity":250}]
            }
        ],
        "currency":{"name":"British pound","internalid":"2"}
    },
    {
        "pricelist":
        [
            {
                "pricelevel":{"name":"Base Price","internalid":"1"},
                "price":[{"price":100.00,"quantitylevel":"1","quantity":0},{"price":95.00,"quan
titylevel":"2","quantity":100},{"price":90.00,"quantitylevel":"3","quantity":150},{"price":85.0
0,"quantitylevel":"4","quantity":200},{"price":80.00,"quantitylevel":"5","quantity":250}]
            },
            {
                "pricelevel":{"name":"Alternate Price 1","internalid":"2"},
                "price":[{"price":99.00,"quantitylevel":"1","quantity":0},{"price":94.00,"quant
itylevel":"2","quantity":100},{"price":89.00,"quantitylevel":"3","quantity":150},{"price":84.00
,"quantitylevel":"4","quantity":200},{"price":79.0","quantitylevel":"5","quantity":250}]
            },
            {
                "pricelevel":{"name":"Alternate Price 2","internalid":"3"},
                "price":[{"price":98.0","quantitylevel":"1","quantity":0},{"price":93.00,"quant
itylevel":"2","quantity":100},{"price":88.00,"quantitylevel":"3","quantity":150},{"price":83.00
,"quantitylevel":"4","quantity":200},{"price":78.00,"quantitylevel":"5","quantity":250}]
            },
            {
                "pricelevel":{"name":"Alternate Price 3","internalid":"4"},
                "price":[{"price":97.00,"quantitylevel":"1","quantity":0},{"price":92.00,"quant
itylevel":"2","quantity":100},{"price":87.00,"quantitylevel":"3","quantity":150},{"price":82.00
,"quantitylevel":"4","quantity":200},{"price":77.00,"quantitylevel":"5","quantity":250}]
            },
            {
                "pricelevel":{"name":"Online Price","internalid":"5"},
                "price":[{"price":96.00,"quantitylevel":"1","quantity":0},{"price":91.00,"quant
itylevel":"2","quantity":100},{"price":86.00,"quantitylevel":"3","quantity":150},{"price":81.00
,"quantitylevel":"4","quantity":200},{"price":76.00,"quantitylevel":"5","quantity":250}]
            }
        ],
        "currency":{"name":"USA","internalid":"1"}
    }
]
```

Sales Order Record Format

JSON

```
{
   "total":64.04,
   "altshippingcost":5.67,
   "taxtotal":4.45,
   "tranid":"120",
   "orderstatus":"E",
   "shipcomplete":false,
   "discounttotal":0.00,
   "entity":"76",
   "billaddress":"Doug Fabre\r\nChess\r\nChess Art Gallery\r\n150 N Ocean Dr\r\nMonterey CA 939
40",
   "salesrep":"-5",
   "ccapproved":false,
   "linkedtrackingnumbers":["1Z6753YA0394527573","1Z6753YA0394249981"],
   "shipmethod":"92",
   "exchangerate":1.00
   "lastmodifieddate":"1/9/2011 11:34 pm",
   "taxrate":"8.25%",
   "id":"769",
   "shipaddresslist":"55",
   "istaxable":true,
   "tobefaxed":false,
   "altsalestotal":0.00,
   "getauth":false,
   "tobeprinted":false,
   "shippingcost":5.67,
   "recordtype":"salesorder",
   "trandate":"10/14/2006",
   "fax":"831-555-5230",
   "customform":"88",
   "links":
   [
      {"trandate":"10/14/2006","tranid":"8","type":"Item Fulfillment","linktype":"Receipt/Fulfi
llment"}
   ],
   "taxitem":"-112",
   "custbody1":"831-555-5229",
   "custbody2":"Do not leave the item outside the house",
   "shipdate":"10/14/2006",
   "createddate":"10/14/2006 2:58 pm",
   "subtotal":53.92,
   "currencyname":"USA",
   "revenuestatus":"A",
   "saleseffectivedate":"10/14/2006",
   "email":chessart@christyscatering.com,
   "item":
   [
      {
         "isclosed":false,"fromjob":false,"amount":8.96,"rate":8.96,"price":"2",
         "istaxable":"T","description":"10 ft Serial Cable DB25M DB25F",
         "custcol6":429,"custcol7":2.5,
         "item":"46","quantity":1,"isestimate":false,"commitinventory":"1",
         "options":
         {
            "CUSTCOL3":792,"CUSTCOL1":4
         }
      },
      {
         "isclosed":false,"fromjob":false,"amount":44.96,"rate":44.96,"price":"2",
         "istaxable":true,
         "item":"80","quantity":1,"isestimate":false,"commitinventory":"1"
      }
   ],
   "excludecommission":false,
   "shipaddress":"Chess\r\nChess Art Gallery\r\n150 N Ocean Dr\r\nMonterey CA 93940","tobeemail
ed":false
}
```

XML

```
<altshippingcost>5.67</altshippingcost>
<total>64.04</total>
<taxtotal>4.45</taxtotal>
<orderstatus>E</orderstatus>
<tranid>120</tranid>
<shipcomplete>F</shipcomplete>
<discounttotal>0.00</discounttotal>
<entity>76</entity>
<billaddress>Doug FabreChessChess Art Gallery150 N Ocean DrMonterey CA 93940</billaddress>
<salesrep>-5</salesrep>
<linkedtrackingnumbers>1Z6753YA0394527573</linkedtrackingnumbers>
<linkedtrackingnumbers>1Z6753YA0394249981</linkedtrackingnumbers>
<ccapproved>F</ccapproved>
<shipmethod>92</shipmethod>
<exchangerate>1.00</exchangerate>
<lastmodifieddate>1/9/2011 11:34 pm</lastmodifieddate>
<taxrate>8.25</taxrate>
<id>769</id>
<shipaddresslist>55</shipaddresslist>
<istaxable>T</istaxable>
<tobefaxed>F</tobefaxed>
<altsalestotal>0.00</altsalestotal>
<getauth>F</getauth>
<tobeprinted>F</tobeprinted>
<shippingcost>5.67</shippingcost>
<recordtype>salesorder</recordtype>
<trandate>10/14/2006</trandate>
<fax>831-555-5230</fax>
<customform>88</customform>
<custbody1>831-555-5229</custbody1>
<custbody2>Do not leave the item outside the house</custbody2>
<shipdate>10/14/2006</shipdate>
<taxitem>-112</taxitem>
<links>
   <trandate>10/14/2006</trandate>
   <tranid>8</tranid>
   <type>Item Fulfillment</type>
   <linktype>Receipt/Fulfillment</linktype>
</links>
<createddate>10/14/2006 2:58 pm</createddate>
<subtotal>53.92</subtotal>
<currencyname>USA</currencyname>
<revenuestatus>A</revenuestatus>
<saleseffectivedate>10/14/2006</saleseffectivedate>
<email>chessart@christyscatering.com</email>
<excludecommission>F</excludecommission>
<item>
     <amount>8.96</amount>
      <fromjob>F</fromjob>
     <isclosed>F</isclosed>
      <price>2</price>
     <rate>8.96</rate>
      <description>10 ft Serial Cable DB25M DB25F</description>
```

```
       <istaxable>T</istaxable>
       <item>46</item>
       <quantity>1</quantity>
       <commitinventory>1</commitinventory>
    <custcol6>429</custcol6>
       <custcol7>2.5</custcol7>
       <isestimate>F</isestimate>
       <options>
             <CUSTCOL3>792</CUSTCOL3>
             <CUSTCOL1>4</CUSTCOL1>
    </options>
</item>
<item>
    <amount>44.96</amount>
      <fromjob>F</fromjob>
       <isclosed>F</isclosed>
       <price>2</price>
       <rate>44.96</rate>
       <istaxable>T</istaxable>
       <item>80</item>
       <quantity>1</quantity>
       <commitinventory>1</commitinventory>
       <isestimate>F</isestimate>
</item>
<shipaddress>ChessChess Art Gallery150 N Ocean DrMonterey CA 93940</shipaddress>
<tobeemailed>F</tobeemailed>
```

RESTlet Status Codes and Error Message Formats

For details about RESTlet errors, see:

- Success Code
- Error Codes
- Notes about RESTlet Errors
- Error Message Formatting
- Error for Incorrect URL

Success Code

RESTlets support the following HTTP success code:

- **200 OK:** The RESTlet request was executed successfully.

  The actual response depends on the request method used. For a GET request, the response contains an entity corresponding to the requested resource. For a POST request the response contains an entity describing or containing the result of the action

Error Codes

RESTlets support the following HTTP error codes:

- **302 Moved Temporarily:** The request was sent to a different data center than the data center in which your company's account resides. When you receive a 302 response, you must recalculate the signature on the request to the correct data center, because the signature is also computed from URL.
- **400 BAD_REQUEST:** The RESTlet request failed with a user error.
- **401 UNAUTHORIZED:** There is not a valid NetSuite login session for the RESTlet calls.
- **403 FORBIDDEN:** RESTlet request sent to invalid domain, meaning a domain other than https://rest.netsuite.com.
- **404 NOT_FOUND:** A RESTlet script is not defined in the RESTlet request.
- **405 METHOD_NOT_ALLOWED:** The RESTlet request method is not valid.
- **415 UNSUPPORTED_MEDIA_TYPE:** An unsupported content type was specified. (Only JSON and text are allowed.)
- **500 INTERNAL_SERVER_ERROR (unexpected errors):** Occurs for non-user errors that cannot be recovered by resubmitting the same request.

  If this type of error occurs, contact Customer Support to file a case.
- **503 SERVICE_UNAVAILABLE:** The NetSuite database is offline or a database connection is not available.

For additional information about HTTP status codes, see http://www.w3schools.com/tags/ref_httpmessages.asp

Notes about RESTlet Errors

- Any errors encountered at run time that are unhandled return a 400 error. If the user code catches the error, a 200 error is returned.
- An unexpected error is returned with an error ID, for example:

```
Code = UNEXPECTED_ERROR
Msg = An unexpected error occurred. Error ID: fevsjhv41tji2juy3le73
```

- An INVALID_REQUEST error is returned due to malformed syntax in the OAuth header. For example, when the signature method, version, or timestamp parameters are rejected.
- An INVALID_LOGIN_ATTEMPT error is returned when the nonce, consumer key, token, or signature in the OAuth header is invalid.
- The DELETE method is not expected to return anything. In this case, the message is returned:

```
    Return was ignored in DELETE operation.
```

- If users specify a content type other than JSON or TEXT, a 415 error is returned with the following message:

```
Invalid content type. You can only use application/json, application/xml or text/plain with RES
Tlets.
```

- If users provide data in a format different from specified type, the following error is returned with one of the following messages:

```
Error code = INVALID_RETURN_DATA_FORMAT
Error message = Invalid data format. You should return TEXT.
Error message = Invalid data format. You should return a JavaScript object.
```

Error Message Formatting

The following examples show RESTlet error message formatting for JSON and text content types.

JSON

```
{
     "error":
   {
      "code":"SSS_INVALID_SCRIPTLET_ID",
      "message":"That Suitelet is invalid, disabled, or no longer exists."
   }
}
```

XML

```
<error>
        <code>SSS_INVALID_SCRIPTLET_ID</code>
        <message>That Suitelet is invalid, disabled, or no longer exists.</message>
</error>
```

Text

```
<error code: SSS_INVALID_SCRIPTLET_ID
error message: That Suitelet is invalid, disabled, or no longer exists.
```

Error for Incorrect URL

If you receive the following error, make sure that the URL is correct and that it points to the correct RESTlet script ID.

```
SSS_INVALID_SCRIPTLET_ID:  That Suitelet is invalid, disabled, or no longer exists.
```

Tracking and Managing RESTlet Activity

If you use token-based authentication, you have the ability to track calls that were made by external applications to RESTlets hosted in your NetSuite account. You can track RESTlet activity by using integration records.

When using token-based authentication (TBA), you create an integration record to represent each external application that calls RESTlets. You use the integration record to generate the consumer key and secret needed by the application to authenticate. You can also use the record to do the following:

- Block requests that reference the record’s consumer key. You can block requests by setting the record’s State field to Blocked.
- Enable or disable token-based authentication for an external application. Note that an integration record can also be used to track web services requests. An additional authentication option on the record, User Credentials, applies only to web services requests. Checking or clearing the User Credentials box has no effect on whether the application can call RESTlets. The User Credentials option affects only web services calls.

Integration records are located at Setup > Integration > Manage Integrations. The record can be accessed only by administrative users.

For full details on using the integration record to set up token-based authentication, see the help topic Token-based Authentication.

For more information on using integration records in conjunction with RESTlets, see the following topics:

- Ownership of Integration Records
- Using the RESTlets Execution Log
- Finding System Notes for an Integration Record
- More Information

Ownership of Integration Records

When you create an integration record, it is automatically available to you in your NetSuite account. Your NetSuite account is considered to be the owner of the integration record, and the record is fully editable by administrators in your account.

You can also install records in your account that were created elsewhere. For example, an integration record could be bundled, distributed, and installed outside the account where it was created. If you install a bundle that includes an integration record, the record is considered to be an installed record. It is owned by a different NetSuite account. On such records, you can make changes to only two fields: the Note field and the State field. All other fields, including the authentication and Description fields, can be changed only by an authorized user in the account that owns the record. When the owner makes changes to these fields, the new settings are pushed automatically to your account. These changes are not reflected in the system notes that appear in your account.

Using the RESTlets Execution Log

Each integration record includes a subtab labeled RESTlets Execution Log. This log lists RESTlet calls that are uniquely identified with that integration record. That is, the log includes those requests that use token-based authentication and reference the integration record’s consumer key.

Note: Calls made using the NLAuth method of authentication are not logged on any integration record.

For each logged request, the RESTlets Execution Log includes details such as the following:

- The date and time that the call was made.
- The duration of the request.
- The email address of the user who made the request.
- The action taken.

- The corresponding script ID and deployment ID.

Tracking Changes to Integration Records

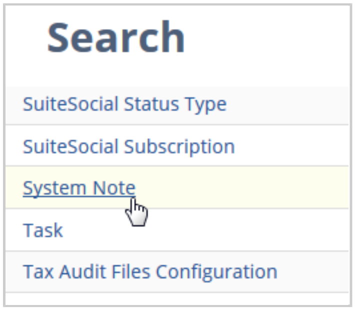
If you want to review changes that were made to an integration record, you can refer to the system notes for that record. System notes are used to track events such as the creation of the record, the initial values of its fields, and subsequent updates. For example, if a user changed the State field from Blocked to Enabled, a system note would provide a record of that change.

For each event, the system records details such as the ID of the user who made the change and the timestamp of the change. If a user assigns a value to a field that already had a value, the system note also shows the field's former setting.

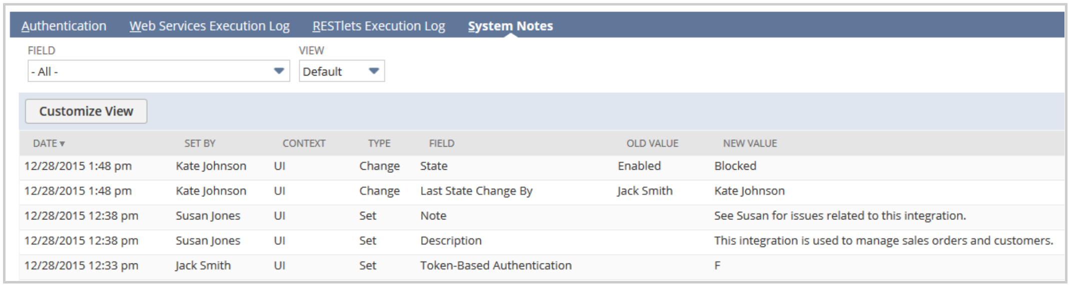
Be aware that in general, system notes are created only for those fields that you are permitted to change. For additional details, see the help topic Special Cases Related to System Notes Logging. Be aware that this section is part of the SuiteTalk (Web Services) Platform Guide. Some of the detail in this guide pertains only or primarily to web services.

You can locate system notes for integration records in either of the following ways:

- By using the System Note search type, at Reports > New Search.



- By clicking on the System Notes subtab of any integration record.



FIELD: - All -
VIEW: Default

DATE ▾	SET BY	CONTEXT	TYPE	FIELD	OLD VALUE	NEW VALUE
12/28/2015 1:48 pm	Kate Johnson	UI	Change	State	Enabled	Blocked
12/28/2015 1:48 pm	Kate Johnson	UI	Change	Last State Change By	Jack Smith	Kate Johnson
12/28/2015 12:38 pm	Susan Jones	UI	Set	Note		See Susan for issues related to this integration.
12/28/2015 12:38 pm	Susan Jones	UI	Set	Description		This integration is used to manage sales orders and customers.
12/28/2015 12:33 pm	Jack Smith	UI	Set	Token-Based Authentication		F

More Information

For more information about managing integration records, see the help topic Managing Integrations, which is part of the SuiteTalk (Web Services) Platform Guide. Be aware that some of the detail in this guide pertains only or primarily to web services:

- Adding an Integration Record
- Regenerating a Consumer Key and Secret
- Distributing by Bundling

- Using Integration Records in Sandbox Accounts
- Removing Integration Records
- Special Cases Related to System Notes Logging