SuiteTalk REST Web Services Records Guide
should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

**Sample Code**

Oracle may provide sample code in SuiteAnswers, the Help Center, User Guides, or elsewhere through help links. All such sample code is provided "as is" and "as available", for use only with an authorized NetSuite Service account, and is made available as a SuiteCloud Technology subject to the SuiteCloud Terms of Service at www.netsuite.com/tos.

Oracle may modify or remove sample code at any time without notice.

**No Excessive Use of the Service**

As the Service is a multi-tenant service offering on shared databases, Customer may not use the Service in excess of limits or thresholds that Oracle considers commercially reasonable for the Service. If Oracle reasonably concludes that a Customer's use is excessive and/or will cause immediate or ongoing performance issues for one or more of Oracle's other customers, Oracle may slow down or throttle Customer's excess use until such time that Customer's use stays within reasonable limits. If Customer's particular usage pattern requires a higher limit or threshold, then the Customer should procure a subscription to the Service that accommodates a higher limit and/or threshold that more effectively aligns with the Customer's actual usage pattern.

**Beta Features**

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are “commercial computer software” or “commercial computer software documentation” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any
operating system, integrated software, any programs embedded, installed or activated on delivered
hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other
Oracle data, is subject to the rights and limitations specified in the license contained in the applicable
contract. The terms governing the U.S. Government’s use of Oracle cloud services are defined by the
applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management
applications. It is not developed or intended for use in any inherently dangerous applications, including
applications that may create a risk of personal injury. If you use this software or hardware in dangerous
applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other
measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages
caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks
of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks
are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD,
Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a
registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content,
products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and
expressly disclaim all warranties of any kind with respect to third-party content, products, and services
unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and
its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use
of third-party content, products, or services, except as set forth in an applicable agreement between you
and Oracle.

This documentation is in pre-General Availability status and is intended for demonstration and preliminary
use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation
and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to
this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of
this documentation.

The information contained in this document is for informational sharing purposes only and should be
considered in your capacity as a customer advisory board member or pursuant to your pre-General
Availability trial agreement only. It is not a commitment to deliver any material, code, or functionality, and
should not be relied upon in making purchasing decisions. The development, release, and timing of any
features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the
exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms
and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle
PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has
been executed by you and Oracle and with which you agree to comply. This document and information
contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle
without prior written consent of Oracle. This document is not part of your license agreement nor can it be
incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Send Us Your Feedback

We'd like to hear your feedback on this document.

Answering the following questions will help us improve our help content:

- Did you find the information you needed? If not, what was missing?
- Did you find any errors?
- Is the information clear?
- Are the examples correct?
- Do you need more examples?
- What did you like most about this document?

Click here to send us your comments. If possible, please provide a page number or section title to identify the content you're describing.

To report software issues, contact NetSuite Customer Support.
## Table of Contents

REST Web Services Available Records ......................................................................................... 1
Account – AssemblyUnbuild ........................................................................................................ 1
BillingAccount – BulkOwnershipTransfer ..................................................................................... 2
CalendarEvent – CustomerSubsidiaryRelationship ..................................................................... 2
Department – DownloadItem ....................................................................................................... 3
EmailTemplate – ExpenseReport ................................................................................................. 4
FairValuePrice – FulfillmentRequest ............................................................................................ 4
GenericResource – GlobalInventoryRelationship ......................................................................... 4
InboundShipment – ItemSupplyPlan ............................................................................................. 5
Job – JournalEntry ..................................................................................................................... 6
KitItem – KitItem ......................................................................................................................... 6
LaborBasedProjectRevenueRule – Location .................................................................................. 6
ManufacturingCostTemplate – MfgPlannedTime ......................................................................... 6
Nexus – NoteType ....................................................................................................................... 7
Opportunity – OtherNameCategory ............................................................................................. 7
Partner – PurchaseRequisition ..................................................................................................... 7
ReceiveInboundShipment – RevRecTemplate ............................................................................... 8
SalesOrder – SupportCase .......................................................................................................... 9
Task – TransferOrder ................................................................................................................. 9
UnitsType – Usage ....................................................................................................................... 10
Vendor – VendorSubsidiaryRelationship .................................................................................... 10
WebSite – Workplace .................................................................................................................. 11

REST Web Services Supported Records ...................................................................................... 12
Billing Account .......................................................................................................................... 12
Charge ....................................................................................................................................... 13
Contact ...................................................................................................................................... 14
Contact Category ..................................................................................................................... 14
Contact Role ........................................................................................................................... 15
Email Template .......................................................................................................................... 16
Journal Entry ............................................................................................................................ 17
Price Book .................................................................................................................................. 18
Price Plan ................................................................................................................................... 19
Subscription .............................................................................................................................. 20
Subscription Line ....................................................................................................................... 22
Subscription Plan ........................................................................................................................ 23
Subsidiary .................................................................................................................................... 24
Usage ......................................................................................................................................... 25

REST Web Services Tutorials ....................................................................................................... 27
SuiteBilling Use Cases ............................................................................................................... 27
Use Case: Manage Your Subscription Catalog ......................................................................... 27
Use Case: Manage Your Subscription Sales .............................................................................. 36
REST Web Services Available Records

This reference document contains a list of records accessible through REST web services. The list includes both generally available and beta records.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

Click the link for each letter to access details about the records in alphabetical order (available fields, sublists, subrecords) and about the operations you can perform on the record.

For additional information about records, see the REST API Browser.

- Account – AssemblyUnbuild
- BillingAccount – BulkOwnershipTransfer
- CalendarEvent – CustomerSubsidiaryRelationship
- Department – DownloadItem
- EmailTemplate – ExpenseReport
- FairValuePrice – FulfillmentRequest
- GenericResource – GlobalInventoryRelationship
- InboundShipment – ItemSupplyPlan
- Job – JournalEntry
- KitItem – KitItem
- LaborBasedProjectRevenueRule – Location
- ManufacturingCostTemplate – MfgPlannedTime
- Nexus – NoteType
- Opportunity – OtherNameCategory
- Partner – PurchaseRequisition
- ReceiveInboundShipment – RevRecTemplate
- SalesOrder – SupportCase
- Task – TransferOrder
- UnitsType – Usage
- Vendor – VendorSubsidiaryRelationship
- WebSite – Workplace

Account – AssemblyUnbuild

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.
The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

### BillingAccount – BulkOwnershipTransfer

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

### CalendarEvent – CustomerSubsidiaryRelationship

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.
The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.
All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- Department – beta – Department
- Deposit – beta – Deposit
- DepositApplication – beta – Deposit Application
- DescriptionItem – beta – Description Item
- DiscountItem – beta – Discount Item
- DownloadItem – beta – Download Item

EmailTemplate – ExpenseReport

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- EmailTemplate – ga – Email Template
- Employee – beta – Employee
- EntityAccountMapping – beta – Entity Account Mapping
- Estimate – beta – Estimate
- ExpenseCategory – beta – Expense Category

FairValuePrice – FulfillmentRequest

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- FairValuePrice – beta – Fair Value Price
- FixedAmountProjectRevenueRule – beta – Fixed Amount Project Revenue Rule
- FulfillmentRequest – beta – Fulfillment Request

GenericResource – GlobalInventoryRelationship

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.
The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- GenericResource – beta – Generic Resource
- GiftCertificateItem – beta – Gift Certificate Item
- GlobalAccountMapping – beta – Global Account Mapping
- GlobalInventoryRelationship – beta – Global Inventory Relationship

InboundShipment – ItemSupplyPlan

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- InboundShipment – beta – Inbound Shipment
- IntercompanyJournalEntry – beta – Intercompany Journal Entry
- IntercompanyTransferOrder – beta – Intercompany Transfer Order
- InventoryAdjustment – beta – Inventory Adjustment
- InventoryCostRevaluation – beta – Inventory Cost Revaluation
- InventoryCount – beta – Inventory Count
- InventoryItem – beta – Inventory Item
- InventoryNumber – beta – Inventory Item Number
- InventoryStatus – beta – Inventory Status
- InventoryStatusChange – beta – Inventory Status Change
- InventoryTransfer – beta – Inventory Transfer
- Invoice – beta – Invoice
- Issue – beta – Issue
- IssueProduct – beta – Issue Product
- ItemAccountMapping – beta – Item Account Mapping
- ItemDemandPlan – beta – Item Demand Plan
- ItemFulfillment – beta – Item Fulfillment
- ItemGroup – beta – Item Group
- ItemLocationConfiguration – beta – Item Location Configuration
- ItemReceipt – beta – Item Receipt
- ItemRevision – beta – Item Revision
- ItemSupplyPlan – beta – Item Supply Plan
Job – JournalEntry

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- Job – beta – Job
- JobStatus – beta – Job Status
- JobType – beta – Job Type
- JournalEntry – ga – Journal Entry

KitItem – KitItem

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- KitItem – beta – Kit Item

LaborBasedProjectRevenueRule – Location

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- LaborBasedProjectRevenueRule – beta – Labor Based Project Revenue Rule
- Location – beta – Location

ManufacturingCostTemplate – MfgPlannedTime

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.
The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- Nexus – beta – Nexus
- NonInventoryPurchaseItem – beta – Noninventory Purchase Item
- NonInventoryResaleItem – beta – Noninventory Resale Item
- NonInventorySaleItem – beta – Noninventory Sale Item
- Note – beta – Note
- NoteType – beta – Note Type

Opportunity – OtherNameCategory

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- Opportunity – beta – Opportunity
- OtherChargeItem – beta – Other Charge Item
- OtherName – beta – Other Name
- OtherNameCategory – beta – Other Name Category

Partner – PurchaseRequisition

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.
The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- Partner – beta – Partner
- PartnerCategory – beta – Partner Category
- Paycheck – beta – Paycheck
- PaycheckJournal – beta – Paycheck Journal
- PaymentItem – beta – Payment Item
- PaymentMethod – beta – Payment Method
- PayrollItem – beta – Payroll Item
- PctCompleteProjectRevenueRule – beta – Pct Complete Project Revenue Rule
- PeriodEndJournal – beta – Period End Journal
- PhoneCall – beta – Phone Call
- PlannedOrder – beta – Planned Order
- PlanningItemCategory – beta – Planning Item Category
- PlanningListGroup – beta – Planning Item Group
- PlanningRuleGroup – beta – Planning Rule Group
- PriceBook – ga – Price Book
- PriceLevel – beta – Price Level
- PricePlan – ga – Price Plan
- PricingGroup – beta – Pricing Group
- ProjectExpenseType – beta – Project Expense Type
- ProjectTask – beta – Project Task
- ProjectTemplate – beta – Project Template
- PromotionCode – beta – Promotion Code
- PurchaseContract – beta – Purchase Contract
- PurchaseOrder – beta – Purchase Order
- PurchaseRequisition – beta – Purchase Requisition

ReceiveInboundShipment – RevRecTemplate

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- ReceiveInboundShipment – beta – Receive Inbound Shipment
- ResourceAllocation – beta – Resource Allocation
The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- SalesOrder – beta – Sales Order
- SalesRole – beta – Sales Role
- SalesTaxItem – beta – Sales Tax Item
- ServiceItem – beta – Service Item
- ShipItem – beta – Ship Item
- Solution – beta – Solution
- StatisticalJournalEntry – beta – Statistical Journal Entry
- StorePickupFulfillment – beta – Store Pickup Fulfillment
- Subscription – ga – Subscription
- SubscriptionChangeOrder – beta – Subscription Change Order
- SubscriptionLine – ga – Subscription Line
- SubscriptionPlan – ga – Subscription Plan
- Subsidiary – ga – Subsidiary
- SubtotalItem – beta – Subtotal Item
- SupplyChainSnapshot – beta – Supply Chain Snapshot
- SupplyChainSnapshotSimulation – beta – Supply Chain Snapshot Simulation
- SupplyChangeOrder – beta – Supply Change Order
- SupplyPlanDefinition – beta – Supply Plan Definition
- SupportCase – beta – Support Case

**Task – TransferOrder**

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.
All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- Task – beta – Task
- TaxAcct – beta – Tax Acct
- TaxGroup – beta – Tax Group
- TaxPeriod – beta – Tax Period
- TaxType – beta – Tax Type
- Term – beta – Term
- TimeBill – beta – Time Bill
- TimeEntry – beta – Time Entry
- TimeSheet – beta – Time Sheet
- Topic – beta – Topic
- TransferOrder – beta – Transfer Order

UnitsType – Usage

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- UnitsType – beta – Units Type
- Usage – ga – Usage

Vendor – VendorSubsidiaryRelationship

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- Vendor – beta – Vendor
- VendorBill – beta – Vendor Bill
- VendorCategory – beta – Vendor Category
- VendorCredit – beta – Vendor Credit
- VendorPayment – beta – Vendor Payment
- VendorReturnAuthorization – beta – Vendor Return Authorization
- VendorSubsidiaryRelationship – beta – Vendor Subsidiary Relationship
WebSite – Workplace

The following section lists records, their support level (generally available or beta), and a link to the REST Records Browser where you can find detailed information about each record.

The majority of SuiteScript records is available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

All records in REST web services are considered beta until indicated otherwise. Beta records are marked with a beta label in the REST API Browser.

- WebSite – beta – Website
- WorkOrder – beta – Work Order
- WorkOrderClose – beta – Work Order Close
- WorkOrderCompletion – beta – Work Order Completion
- WorkOrderIssue – beta – Work Order Issue
- Workplace – beta – Workplace
REST Web Services Supported Records

Most SuiteScript records are available as beta records. Refer to the REST web services API and the developer documentation to learn the status of records.

The following records are supported in REST web services:

Entity Records
- Contact
- Contact Category
- Contact Role

ERP Records
- Journal Entry
- Subsidiary

Marketing Records
- Email Template

SuiteBilling Records
- Billing Account
- Charge
- Price Book
- Price Plan
- Subscription
- Subscription Line
- Subscription Plan
- Usage

Billing Account

A billing account record exposes a SuiteBilling account to REST web services. You can enable Billing Accounts without enabling SuiteBilling.

This record:
- is not a subrecord
- has no subrecords

All elements on this record are accessible through REST web services.

The REST API Browser includes information about the field names and field types of the billing account record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser’s billing account reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for the billing account REST record is billingaccount.
Prerequisites
You must Enabling SuiteBilling Features before you can use this record through REST web services.

Limitations
A billing account REST record does not show related records.

Code Sample

```json
{
  "startDate": "2019-2-20",
  "customer": {"id": 1},
  "billingSchedule": {"id": 1},
  "currency": {"id": 1}
}
```

Charge
A charge record exposes a SuiteBilling charge to REST web services. This record:

- is not a subrecord
- has no subrecords

The REST API Browser includes information about the field names and field types of the charge record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser's charge reference page.

Record ID
The record ID for the charge REST record is `charge`.

Prerequisites
You must Enabling SuiteBilling Features before you can use this record through REST web services.

Additional Details
You must specify the stage field as an enumeration string. Possible values can be obtained through the Metadata Catalog. Charge type must be specified as an internal ID. You can obtain possible values via SuiteScript while creating a charge through the User Interface.

Code Sample
This sample shows a common billing usage case.

```bash
POST https://demo123.suitetalk.api.netuite.com/services/rest/record/v1/charge
```
Contact

A contact record exposes a contact to REST web services.

This record:
- is not a subrecord
- has no subrecords

The REST API Browser includes information about the field names and field types of the contact record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser's contact reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for a contact REST record is `contact`.

Code Sample

In the following example, `<accountID>` represents your account ID.

HTTP request: https://<accountID>/services/rest/record/v1/contact/973

```

<table>
<thead>
<tr>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>{</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
</tbody>
</table>
|10 | }
```

Contact Category

A contact category record exposes a contact category to REST web services.

This record:
- is not a subrecord
- has no subrecords

All elements on this record are accessible through REST web services.

Record actions supported for this record type include delete, submitnew, and submitter. For more information, see the help topic Supported Record Actions.
The REST API Browser includes information about the field names and field types of the contact category record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser’s contactCategory reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for a contact category REST record is contactcategory.

Code Sample

In the following example, <accountID> represents your account ID.

HTTP request: https://<accountID>/services/rest/record/v1/contactcategory/4

```json
{
  "name": "ContactCategoryFour",
  "externalid": "3929345",
  "private": true
}
```

Contact Role

A contact role record exposes a contact role to REST web services.

This record:
- is not a subrecord
- has no subrecords

All elements on this record are accessible through REST web services.

The REST API Browser includes information about the field names and field types of the contact role record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser’s contactRole reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for a contact role REST record is contactrole.

Code Sample

In the following example, <accountID> represents your account ID.

HTTP request: https://<accountID>/services/rest/record/v1/contactrole/1

```json
{
  "name": "ContactRoleOne",
  "description": "Key Contact",
}
```
Email Template

An email template record exposes an email template to REST web services.

This record is not a subrecord.

The REST API Browser includes information about the field names and field types of the email template record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser’s emailTemplate reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for the email template REST record is emailtemplate.

The record ID for the email template REST subrecord is CrmTemplateCategorySubsidiary.

Prerequisites

You must enable the CRM feature before you can use this record through REST web services.

Exposed Elements

The following email template record elements are exposed to REST web services:

- {addcompanyaddress} – add company address
- {addunsubscribelink} – add unsubscribe link
- {content} – content
- {description} – description
- {id} – ID
- {isautoconverted} – is automatically converted
- {isinactive} – is inactive
- {isprivate} – is private
- {mediaitem} – media item
- {name} – name
- {recordtype} – record type
- {subject} – subject
- {subscription} – subscription

Code Sample

In the following example, <accountID> represents your account ID.
HTTP request: https://<accountID>/services/rest/record/v1/emailtemplate/973

```json
{
  "name": "EmailTemplateNine",
  "addunsubscribelink": true,
  "description": "Marketing Campaign Template"
}
```

Journal Entry

You use the journal entry record to adjust balances in accounts. Journal entries let you change the value of any set of accounts without having to enter a posting transaction. In an account that has the Multi-Book Accounting feature enabled, you can also use this record to create book specific journal entries.

For more information on journal entries in NetSuite, see the help topic Journal Entries.

This record:

- is not a subrecord
- has the following subrecords:
  - journalentryline
  - accountingbookdetail

The REST API Browser includes information about the field names and field types of the journal entry record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser’s journalentry reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for the journal entry REST record is journalentry.

Prerequisites

There are no prerequisites for using the record in REST.

Limitations

- Tax-related fields are not supported in REST.
- The accounting book detail sublist is part of the Multi-Book Accounting feature (and its related features). For more information, see the help topic Multi-Book Accounting.
- Read-only sublists, including system notes and GL impact, are not supported in REST.

Code Sample

```json
{
  "subsidiary": { "id": "1" },
}
```
Price Book


This record:

- is not a subrecord
- has no subrecords

The REST API browser includes information about the field names and field types of the price book record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser's price book reference page.

For information on using the REST API browser, see the help topic The REST API Browser.

Record ID

The record ID for a price book REST record is `pricebook`.

Prerequisites

You must Enabling SuiteBilling Features before you can use this record through REST web services.

Code Samples

These samples show common use cases for creating price books with all header fields.
Price Plan

A price plan record exposes a SuiteBilling price plan to REST web services.

This record:
- is not a subrecord
- has no subrecords

The REST API Browser includes information about the field names and field types of the price plan record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser’s price plan reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for a price plan REST record is priceplan.

Prerequisites

You must Enabling SuiteBilling Features before you can use this record through REST web services.

Code Sample
Subscription

A subscription record exposes a SuiteBilling subscription to REST web services. This record:

- is not a subrecord
- has no subrecords

The REST API Browser includes information about the field names and field types of the subscription record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser's subscription reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for a subscription REST record is subscription.

Prerequisites

You must Enabling SuiteBilling Features before you can use this record through REST web services.

Limitations

A subscription REST record does not show change orders, related records, or system information.

Code Samples

These samples show common use cases for creating subscriptions.

Create with all Header Fields

```json
{
    "name": "REST-Subscription",
    "customer": 4,
    "billingAccount": 1,
    "subscriptionPlan": 8,
    "priceBook": 1,
    "initialTerm": {"id": -102},
    "startDate": "2019-2-20",
    "billingSchedule": 1,
    "billingFrequency": "MONTHLY",
    "defaultRenewalTerm": {"id": 2},
    "autoRenewal": true,
    "advanceRenewalPeriodNumber": 5,
    "advanceRenewalPeriodUnit": "DAYS",
    "defaultRenewalMethod": "CREATE_NEW_SUBSCRIPTION",
    "defaultRenewalPriceBook": {"id": 103},
    "defaultRenewalPlan": {"id": 8},
    "defaultRenewalTranType": "SalesOrd",
    "endDate": "2022-3-5"
}
```
Create with Different Pricing Values

```
{
  "name": "REST-Subscription_with_pricing",
  "customer": 4,
  "billingAccount": 1,
  "subscriptionPlan": 8,
  "priceBook": 1,
  "initialTerm": {
    "id": 1
  },
  "startDate": "2019-2-20",
  "billingFrequency": "MONTHLY",
  "priceinterval": {
    "Items": [
      {
        "subscriptionplanlinenumber": 2,
        "quantity": 3,
        "repeatEvery": 2,
        "pricePlan": {
          "id": 828
        },
        "frequency": "ANNUALLY"
      }
    ]
  }
}
```

Create a New Price Interval

```
{
  "name": "REST-Subscription_with_pricing",
  "customer": 4,
  "billingAccount": 1,
  "subscriptionPlan": 8,
  "priceBook": 1,
  "initialTerm": {
    "id": 1
  },
  "startDate": "2019-2-20",
  "billingFrequency": "MONTHLY",
  "priceinterval": {
    "Items": [
      {
        "subscriptionplanlinenumber": 2,
        "linenumber": 2,
        "quantity": 2,
        "startDate": "2019-4-20",
        "pricePlan": {
          "id": 1028
        },
        "frequency": "MONTHLY",
        "repeatEvery": 1
      }
    ]
  }
}
```

Create with Different Subline Values

```
{
  "name": "REST-Subscription_with_pricing",
  "customer": 4,
  "billingAccount": 1,
  "subscriptionPlan": 8,
  "priceBook": 1,
  "initialTerm": {
    "id": 1
  },
  "startDate": "2019-2-20",
  "billingFrequency": "MONTHLY",
  "subscriptionline": {
    "Items": [
      {
        "linenumber": 1
      }
    ]
  }
}
```
Create with an Add-On Item

Subscription Line

A subscription line record exposes a SuiteBilling subscription line to REST web services.

This record is not a subrecord, but is always tied to a subscription.

The REST API Browser includes information about the field names and field types of the subscription line record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser’s subscription line reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for a subscription line REST record is subscriptionline.
Prerequisites

You must Enabling SuiteBilling Features before you can use this record through REST web services.

Limitations

A subscription line REST record does not show system notes.

You cannot directly create or delete the exposed subscription line record, but you can update some of the fields.

Code Sample

```
Sample Update:
{
  "subscriptionlinestatus": "PENDING_ACTIVATION",
  "proratestartdate": false,
  "prorateenddate": false,
  "includeinrenewal": false
}
```

Subscription Plan

A subscription plan is a stand-alone record that exposes a SuiteBilling subscription plan to REST web services.

A subscription requires a subscription plan, but the subscription plan is not always tied to a subscription.

The REST API Browser includes information about the field names and field types of the subscription plan record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser's subscription plan reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for a subscription plan REST record is subscriptionplan.

The record ID for a subscription plan REST subrecord is subscriptionplanmember.

Prerequisites

You must Enabling SuiteBilling Features before you can use this record through REST web services.

Limitations

A subscription plan REST record does not show price books or related records.

When you access this record through REST web services, there is no link available for creating price books.
Code Sample

This sample shows a common use case for creating a subscription plan.

Create with all Header Fields

```json
{
  "itemId": "NotRested-2",
  "initialTerm": {
    "id": 1
  },
  "member": {
    "items": [{
      "item": {
        "id": 5
      },
      "isrequired": true,
      "subscriptionlinetype": "2",
      "renewaloption": "DIFFERENT_PLAN"
    }]
  }
}
```

Subsidiary

NetSuite exposes the subsidiary record to REST web services. The subsidiary record enables you to manage data for a hierarchical structure of separate legal entities. See the help topic Subsidiaries in OneWorld.

The REST API Browser includes information about the field names and field types of the subsidiary record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser's subsidiary reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.

Record ID

The record ID for the subsidiary REST record is subsidiary.

The record IDs for the subsidiary REST subrecords are:

- mainaddress
- shippingaddress
- returnaddress

Prerequisites

You must either use NetSuite OneWorld or have the Subsidiaries hidden feature enabled before you can use this record through REST web services.

Exposed Elements

The following subsidiary record elements are exposed to REST web services:

- (internalid) - internal id
- (externalid) - external id
- (isinactive) - subsidiary is inactive
- (legalname) - legal name
- (name) - name
- (fullname) - (search-only field)
- (parent) - subsubsidiary of
- (currency) - currency
- (iselimination) - elimination
- (country) - country
- (state) - state/province
- (url) - web site
- (email) - return email address
- (fax) - fax

The following subsidiary subrecord elements are exposed to REST web services:
- (mainaddress) - main address
- (shippingaddress) - shipping address
- (returnaddress) - return address

Limitations

Only GET (Read, Search) operation is supported for REST web services. POST (create), PATCH (update), and DELETE are not supported.

Code Samples

The following code samples show a common use case for getting supported elements from the subsidiary REST record.

```
1. GET /services/rest/record/v1/subsidiary/
```

```
3. GET /services/rest/record/v1/subsidiary?q=fullName CONTAIN 'parent'
```

Usage

A usage record exposes SuiteBilling usage to REST web services. This record:
- is not a subrecord
- has no subrecords

The REST API Browser includes information about the field names and field types of the usage record, and about the HTTP methods, request parameters, and operations available to this record. For details, see the REST API Browser's usage reference page.

For information on using the REST API Browser, see the help topic The REST API Browser.
Record ID

The record ID for a usage REST record is `usage`.

Prerequisites

You must Enabling SuiteBilling Features before you can use this record through REST web services.

This record also requires, at a minimum, a customer, subscription, subscription plan, and subscription line.

Code Sample

This sample shows a common usage case.

```json
POST https://demo123.suitetalk.api.netsuite.com/services/rest/record/v1/usage
{
    "memo": "Record of usage",
    "usageQuantity": 33,
    "usageDate": "2020-07-14",
    "customer": 5,
    "subscriptionPlan": 8,
    "usageSubscription": 208,
    "usageSubscriptionLine": 235
}
```
REST Web Services Tutorials

The following sections contain end-to-end guidelines for specific business scenarios.

- SuiteBilling Use Cases

SuiteBilling Use Cases

This section includes the following sample use cases for managing subscriptions using REST web services:

- Use Case: Manage Your Subscription Catalog
- Use Case: Manage Your Subscription Sales

Use Case: Manage Your Subscription Catalog

REST services provide a convenient channel to manage your subscription catalog.

In this use case, you create the necessary records using REST for your fictitious company, CED Cyber Solutions.

This use case demonstrates how to:

1. Create Service Items
2. Create a Subscription Plan
3. Create a Price Plan
4. Create a Price Book
5. Retrieve Records

Create Service Items

Before you begin, you need to create service items for your subscription plan. The security subscription plan that you want to sell includes the following items:

- Setup
- License
- Seats
- Bronze Support
- Silver Support
- Gold Support

You need to create a subscription term because you want your subscription plan to have an initial term of three years.

Although the subscription plan depends on these records, this use case does not cover how to create them through REST. For now, you can create them through the user interface.
After you create each record, make note of its internal ID so that you can reference it. You can find the internal ID of each record by navigating to the record's list page.

The following table lists prerequisite records along with the record type and example internal ID for each:

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Record Name</th>
<th>Example Internal ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Items</td>
<td>License</td>
<td>1</td>
</tr>
<tr>
<td>Service Items</td>
<td>Setup</td>
<td>2</td>
</tr>
<tr>
<td>Service Items</td>
<td>Seats</td>
<td>3</td>
</tr>
<tr>
<td>Service Items</td>
<td>Bronze Support</td>
<td>4</td>
</tr>
<tr>
<td>Service Items</td>
<td>Silver Support</td>
<td>5</td>
</tr>
<tr>
<td>Service Items</td>
<td>Gold Support</td>
<td>6</td>
</tr>
<tr>
<td>Subscription Term</td>
<td>Three Years</td>
<td>1</td>
</tr>
</tbody>
</table>

Create a Subscription Plan

To create records, you send POST requests to the endpoint for the record type. The request body should contain information for all required fields and any optional fields.

Your requirements are as follows:

- The plan name is "DEFEND Package"
- Setup is a one-time line
- Licensing and the number of users (seats) are recurring lines
- Customers must choose one of the three support tiers

**Note:** At present time, there is no way to enforce this behavior through NetSuite, so you can make the lines optional and let the sales person enforce this rule.

- All recurring lines should be prorated except for seats
- All lines are billed in advance

To set the initialTerm to the subscription term that you have created, you need to reference an existing record. Because you created a custom subscription term, wrap the ID of that term in an object with the ID property. Wrapping the ID in an object is commonly how records are referenced in REST.

Lines on subscription plans are specified in the member sublist. The member sublist is an object with an items property. The items property is an array of objects that each represent a line. The lines have their own fields that must be defined. Although the lineNumber can be inferred in some cases, it is best to always specify it. Lines are 1-indexed. This is the common structure for sublists in requests.

Some fields on the lines in the member sublist require enumeration values. Usually, the possible values are clearly defined in the Metadata Catalog. However, sometimes they can be unclear. For example, subscriptionLineType has an enumeration value of integer strings. Those strings map to subscription line types:

- 1 – One-Time
- 2 – Recurring
3 – Usage

A successful request returns a response with a HTTP status of **204 No Content**. In the response headers, there is a Location key with an associated value representing the endpoint at which you can access your newly created record. Make note of the ID so that you can reference your subscription plan later. The ID for your example subscription plan is 1.

### Example Subscription Plan

```json
{
  "itemId": "DEFEND Package",
  "initialTerm": { "id": "1" },
  "member": {
    "items": [
      { "lineNumber": 1,
        "item": { "id": "1" },
        "isRequired": true,
        "subscriptionLineType": "1",
        "billingMode": "IN_ADVANCE",
        "renewalOption": "DIFFERENT_PLAN"
      },
      { "lineNumber": 2,
        "item": { "id": "2" },
        "isRequired": true,
        "subscriptionLineType": "2",
        "prorateStartDate": true,
        "prorateEndDate": true,
        "billingMode": "IN_ADVANCE",
        "renewalOption": "ALWAYS"
      },
      { "lineNumber": 3,
        "item": { "id": "3" },
        "isRequired": true,
        "subscriptionLineType": "2",
        "prorateStartDate": false,
        "prorateEndDate": false,
        "billingMode": "IN_ADVANCE",
        "renewalOption": "ALWAYS"
      },
      { "lineNumber": 4,
        "item": { "id": "4" },
        "isRequired": false,
        "subscriptionLineType": "2",
        "prorateStartDate": true,
        "prorateEndDate": true,
        "billingMode": "IN_ADVANCE",
        "renewalOption": "ALWAYS"
      },
      { "lineNumber": 5,
        "item": { "id": "5" },
        "isRequired": false,
        "subscriptionLineType": "2",
        "prorateStartDate": true,
        "prorateEndDate": true,
        "billingMode": "IN_ADVANCE",
        "renewalOption": "ALWAYS"
      },
      { "lineNumber": 6,
        "item": { "id": "6" },
        "isRequired": false,
        "subscriptionLineType": "2",
        "prorateStartDate": true,
        "prorateEndDate": true,
        "billingMode": "IN_ADVANCE",
        "renewalOption": "ALWAYS"
      }
    ]
  }
}
```
Create a Price Plan

Your pricing requirements are as follows:

- Setup costs $100.00
- A license costs $1000.00
- Seats have a volume-based cost that decreases as your customer adds more users
  - For 0 - 20 seats, each seat costs $7.00
  - For 21 - 50 seats, each seat costs $6.00
  - For 51+ seats, each seat costs $5.00
- Support prices change over time
  - Bronze starts at $30.00 and drops to $20.00 after the first year
  - Silver starts at $60.00 and drops to $40.00 after the first year
  - Gold starts at $90.00 and drops to $60.00 after the first year

There are six lines on your subscription plan, which means that you need at least six price plans. You need three more price plans to account for the price drop after the first year on your support lines. To satisfy these requirements, you need nine price plans.

Each price plan requires its own request.

**Price Plan 1**

```json
{
  "currency": { "id": "1" },
  "pricePlanType": "2",
  "priceTiers": {
    "items": [
      { "fromVal": 0,
      "pricingOption": { "id": ":102" },
      "value": 100.00
      }
    ]
  }
}
```

**Price Plan 2**

```json
{
  "currency": { "id": "1" },
  "pricePlanType": "2",
  "priceTiers": {
    "items": [
      { "fromVal": 0,
      "pricingOption": { "id": ":102" },
      "value": 250.00
      }
    ]
  }
}
```
Price Plan 3

```json
{
  "currency": { "id": "1" },
  "pricePlanType": "4",
  "priceTiers": [
    { "fromVal": 0,
      "pricingOption": { "id": "-101" },
      "value": 7.00
    },
    { "fromVal": 20,
      "pricingOption": { "id": "-101" },
      "value": 6.00
    },
    { "fromVal": 50,
      "pricingOption": { "id": "-101" },
      "value": 5.00
    }
  ]
}
```

Price Plan 4

```json
{
  "currency": { "id": "1" },
  "pricePlanType": "2",
  "priceTiers": [
    { "fromVal": 0,
      "pricingOption": { "id": "-102" },
      "value": 30.00
    }
  ]
}
```

Price Plan 5

```json
{
  "currency": { "id": "1" },
  "pricePlanType": "2",
  "priceTiers": [
    { "fromVal": 0,
      "pricingOption": { "id": "-102" },
      "value": 20.00
    }
  ]
}
```
Use Case: Manage Your Subscription Catalog

Price Plan 6

```json
{
  "currency": { "id": 1 },
  "pricePlanType": "2",
  "priceTiers": {
    "items": [
      {
        "fromVal": 0,
        "pricingOption": { "id": "-102" },
        "value": 60.00
      }
    ]
  }
}
```

Price Plan 7

```json
{
  "currency": { "id": 1 },
  "pricePlanType": "2",
  "priceTiers": {
    "items": [
      {
        "fromVal": 0,
        "pricingOption": { "id": "-102" },
        "value": 40.00
      }
    ]
  }
}
```

Price Plan 8

```json
{
  "currency": { "id": 1 },
  "pricePlanType": "2",
  "priceTiers": {
    "items": [
      {
        "fromVal": 0,
        "pricingOption": { "id": "-102" },
        "value": 90.00
      }
    ]
  }
}
```

Price Plan 9

```json
{
  "currency": { "id": 1 },
  "pricePlanType": "2",
  "priceTiers": {
    "items": [
      {
        "fromVal": 0,
        "pricingOption": { "id": "-102" },
        "value": 60.00
      }
    ]
  }
}
```
Create a Price Book

Your requirements are as follows:

- Setup is a one-time cost
- License cost is billed annually and prorated by month
- Seats are billed monthly and prorated by day
- Support is billed monthly, prorated by day, and prices decrease after the first year

The subscriptionPlanLineNumber has an integer value and is the number of the line to which the price interval should apply. The sublist line number cannot be inferred by the system because the relationship between subscription plan lines and price intervals is one-to-many.

The repeatEvery is connected to the frequency and determines how often the line should be billed. It has an enum value instead of an integer. The values map directly to their integer values.

For one-time lines, you must specify that repeatEvery is 0 and prorateBy is an empty string.

The startOffsetUnit and startOffsetValue are related to time-based pricing. Combined, the two fields determine when the pricing of a certain line should follow the tiers in the referenced price plan. For example, the support lines (4, 5, and 6) have two price intervals each. The first interval defines the pricing for the first year as startOffsetUnit = YEAR and startOffsetValue = 1. The second interval defines the pricing for the second year as startOffsetUnit = YEAR and startOffsetValue = 2. This results in a reduced recurring price for support in every year after the first.

On a successful response, you have completed the process for creating a subscription plan and price plans linked by a single price book.

```json

Price Book

```

```json
//
// subscriptionPlan: { "id": "1" },
// currency: { "id": "1" },
// name: "TEST Pricing",
// priceInterval: {
//   "items": [
//     { "subscriptionPlanLineNumber": 1,
//       "pricePlan": { "id": "10" },
//       "frequency": "ONETIME",
//       "repeatEvery": "0",
//       "startOffsetUnit": "MONTH",
//       "startOffsetValue": 1,
//       "prorateBy": "" }
//   ],
//   { "subscriptionPlanLineNumber": 2,
//     "pricePlan": { "id": "11" },
//     "frequency": "ANNUALLY",
//     "repeatEvery": "1",
//     "startOffsetUnit": "MONTH",
//     "startOffsetValue": 1,
//     "prorateBy": "MONTH"
//   },
//   { "subscriptionPlanLineNumber": 3,
//     "pricePlan": { "id": "12" },
//     "frequency": "MONTHLY",
//     "repeatEvery": "1",
//     "startOffsetUnit": "MONTH",
//   }
// }
```
Retrieve Records

You can retrieve records using a GET request. You can also use a GET request for validating records after creation. The endpoint is similar to the POST requests. The only difference is that the ID of the record is appended to the end.

Example: In this example, the GET request returns the response body that follows.
Note: The response body is slightly different from the request body because there are other optional fields. These fields were set to their default values internally. The member sublist has its own endpoint that you can access to gather field data for each line.

This GET request:

```
GET http://demo123.suitetalk.api.netsuite.com/services/rest/record/v1/subscriptionplan/1
```

Returns this response body:

```
{
    "autoRenewal": false,
    "createdDate": "2020-1-1T00:00:00Z",
    "customForm": "-950",
    "defaultRenewalTerm": {
        "id": "1",
        "refName": "Three Years",
        "id": "1",
        "includeChildren": false,
        "incomeAccount": {
            "id": "1",
            "refName": "Sales",
            "isInactive": false,
            "itemId": "DEFEND Package",
            "lastModifiedDate": "2020-1-1T00:00:00Z",
            "member": {
                "id": "1",
                "refName": "Three Years"
            }
        }
    },
    "initialTerm": {
        "id": "1",
        "refName": "Three Years"
    },
    "isInactive": false,
    "itemId": "DEFEND Package",
    "lastModifiedDate": "2020-1-1T00:00:00Z",
    "member": {
        "id": "1",
        "refName": "Three Years"
    }
}
```
Use Case: Manage Your Subscription Sales

You can use REST services to manage your subscription sales.

In this use case, you work through the process of using REST to sell a subscription plan for a fictitious company, CED Cyber Solutions.

This use case demonstrates how to:

1. Create Prerequisite Records
2. Create a Billing Account
3. Create a Draft Subscription
4. Edit Your Draft Subscription
5. Add an Add-On Item

Create Prerequisite Records

Before you begin, you need to create the following records:

- **Subscription Plan** – If you have not created a subscription plan (or its associated pricing records) yet and would like to do so using REST, see Managing Your Catalog.
- **Price Plan** – See the help topic Creating Price Plans
- **Price Book** - See the help topic Creating Price Books
- **Customer** – You need a customer to sell your subscription plan to. In this example, your customer is called Clean Water Co.
- **Billing Schedule** – You need to create a charge-based billing schedule to determine how frequently we bill the customer. Your billing schedule is monthly, on the first of the month.

After creating these records, make note of their internal IDs. Throughout this example, use the IDs listed in the following table:

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Name</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscription Plan</td>
<td>DEFEND Package</td>
<td>1</td>
</tr>
<tr>
<td>Price Book</td>
<td>Standard Pricing</td>
<td>1</td>
</tr>
<tr>
<td>Customer</td>
<td>Clean Water Co.</td>
<td>1</td>
</tr>
<tr>
<td>Subscription</td>
<td>Three Years</td>
<td>1</td>
</tr>
<tr>
<td>Billing Schedule</td>
<td>Monthly, 1st of the month</td>
<td>1</td>
</tr>
<tr>
<td>Service Items</td>
<td>Managed Detection and Response</td>
<td>7</td>
</tr>
</tbody>
</table>

Create a Billing Account

Clean Water Co. wants to be billed on the first of the month. They use the same currency as your primary currency. Since they are starting their subscription on January 1, 2020, that is when billing starts.
The **startDate** must always be in the format YYYY-MM-DD. Use this format regardless of the date format preferences set.

A successful request returns a response with HTTP status **204 No Content**. The returned response headers contain the endpoint for the new record. Make note of the Billing Account ID. The example Billing Account internal ID is 1.

### Example Billing Account

```
{
  "billingSchedule": { "id": "1" },
  "currency": { "id": "1" },
  "customer": { "id": "16" },
  "name": "REST Billing Account",
  "startDate": "2020-01-01"
}
```

### Create a Draft Subscription

Your subscription references all records that you have created.

On a successful response, make note of the subscription ID. Use the ID to make changes to your draft subscription. The ID for your example subscription is 1.

### Example Subscription

```
{
  "customer": { "id": "1" },
  "billingAccount": { "id": "1" },
  "subscriptionPlan": { "id": "1" },
  "priceBook": { "id": "1" },
  "initialTerm": { "id": "1" }
}
```

### Edit Your Draft Subscription

When you created your subscription, you chose not to supply information about the line items included or the pricing of those line items. This means that the fields use the default values from the subscription plan.

If desired, you can change the defaults. To change the defaults, you send a PATCH request with the ID of the target record appended to the end of the patch request. The contents of the request body only need to include your changes. Make sure that all changes to dependent fields are included.

Clean Water Co. wants the silver support tier and has 25 users. Because they are one of your first customers, they receive a 50% discount on their setup costs.

**Note:** Leaving a field undefined leaves it unchanged. If you want to set the value of a field to an empty value, you must explicitly set the field to null in the request body.

A successful request returns a response with HTTP status **204 No Content**. The endpoint for the modified record is returned in the response headers.

### Example Edited Subscription

```
```
Add an Add-On Item

Before adding add-on items to any subscription plan, you must Enabling SuiteBilling Features.

Clean Water Co. would like to add Managed Detection and Response to their subscription. The requirements for the line are as follows:

- Bill in advance
- Always add to renewal subscriptions
- Have a prorated start and end date
- Charge $50.00 per month, prorated by day

Adding an add-on item to a draft subscription is an edit. You complete this edit by sending a PATCH request to the subscription endpoint with the ID appended to the end.

Because the add-on item is not tied to any subscription plan or price book, you must supply all of the subscription line and price book line information.

Because the add-on line requires new pricing information, you must create another price plan. If you are unfamiliar with this process, see Use Case: Manage Your Subscription Catalog. You assign an ID of 19 to the new price plan.

If the customer decides that they no longer want the add-on item, you can PATCH the subscription to mark the line as not included.

Example Add-On Items
Use Case: Manage Your Subscription Sales

```json
{
  "priceInterval": {
    "items": [
      {
        "subscriptionPlanLineNumber": 7,
        "pricePlan": { "id": "19" },
        "frequency": "MONTHLY",
        "repeatEvery": "1",
        "startOffsetUnit": "MONTH",
        "startOffsetValue": 1,
        "prorateBy": "DAY"
      }
    ]
  }
}
```