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**

Oracle may make available to Customer certain features that are labeled “beta” that are not yet generally available. To use such features, Customer acknowledges and agrees that such beta features are subject to the terms and conditions accepted by Customer upon activation of the feature, or in the absence of such terms, subject to the limitations for the feature described in the User Guide and as follows: The beta feature is a prototype or beta version only and is not error or bug free and Customer agrees that it will use the beta feature carefully and will not use it in any way which might result in any loss, corruption or unauthorized access of or to its or any third party's property or information. Customer must promptly report to Oracle any defects, errors or other problems in beta features to support@netsuite.com or other designated contact for the specific beta feature. Oracle cannot guarantee the continued availability of such beta features and may substantially modify or cease providing such beta features without entitling Customer to any refund, credit, or other compensation. Oracle makes no representations or warranties regarding functionality or use of beta features and Oracle shall have no liability for any lost data, incomplete data, re-run time, inaccurate input, work delay, lost profits or adverse effect on the performance of the Service resulting from the use of beta features. Oracle's standard service levels, warranties and related commitments regarding the Service shall not apply to beta features and they may not be fully supported by Oracle's customer support. These limitations and exclusions shall apply until the date that Oracle at its sole option makes a beta feature generally available to its customers and partners as part of the Service without a “beta” label.
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

List of REST Web Services Records ............................................................................................................. 1
  A ................................................................................................................................................................. 1
  B ................................................................................................................................................................. 2
  C ................................................................................................................................................................. 2
  D ................................................................................................................................................................. 3
  E ................................................................................................................................................................. 4
  F ................................................................................................................................................................. 4
  G ................................................................................................................................................................. 4
  I ................................................................................................................................................................. 4
  J ................................................................................................................................................................. 5
  K ................................................................................................................................................................. 5
  L ................................................................................................................................................................. 6
  M ................................................................................................................................................................. 6
  N ................................................................................................................................................................. 6
  O ................................................................................................................................................................. 7
  P ................................................................................................................................................................. 7
  R ................................................................................................................................................................. 8
  S ................................................................................................................................................................. 8
  T ................................................................................................................................................................. 9
  U ................................................................................................................................................................. 9
  V ................................................................................................................................................................. 9
  W ................................................................................................................................................................. 10

REST Web Services Supported Records ........................................................................................................ 11
  Billing Account .............................................................................................................................................. 11
  Price Book .................................................................................................................................................... 12
  Price Plan ..................................................................................................................................................... 13
  Subsidiary ..................................................................................................................................................... 19

REST Web Services Tutorials .......................................................................................................................... 21
  SuiteBilling Use Cases ................................................................................................................................. 21
  Use Case: Manage Your Subscription Catalog ............................................................................................ 21
  Use Case: Manage Your Subscription Sales .................................................................................................. 31
List of REST Web Services Records

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

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

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.

- A
- B
- C
- D
- E
- F
- G
- I
- J
- K
- L
- M
- N
- O
- P
- R
- S
- T
- U
- V
- W

A

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- Account – beta – Account
- AccountingBook – beta – Accounting Book
- AccountingContext – beta – Accounting Context
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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- **AccountingPeriod** – beta – Accounting Period
- **AdvIntercompanyJournalEntry** – beta – Advanced Intercompany Journal Entry
- **AllocationSchedule** – beta – Allocation Schedule
- **AmortizationSchedule** – beta – Amortization Schedule
- **AmortizationTemplate** – beta – Amortization Template
- **AssemblyBuild** – beta – Assembly Build
- **AssemblyItem** – beta – Assembly Item
- **AssemblyUnbuild** – beta – Assembly Unbuild

B

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- **BillingAccount** – beta – Billing Account
- **BillingClass** – beta – Billing Class
- **BillingRateCard** – beta – Billing Rate Card
- **BillingRevenueEvent** – beta – Billing Revenue Event
- **BillingSchedule** – beta – Billing Schedule
- **Bin** – beta – Bin
- **BinTransfer** – beta – Bin Transfer
- **BinWorksheet** – beta – Bin Worksheet
- **BlanketPurchaseOrder** – beta – Blanket Purchase Order
- **Bom** – beta – Bom
- **BomRevision** – beta – Bom Revision
- **BudgetExchangeRate** – beta – Budget Exchange Rate
- **BulkOwnershipTransfer** – beta – Bulk Ownership Transfer
- **BundleInstallationScript** – beta – Bundle Installation Script

C

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- **CalendarEvent** – beta – Calendar Event
- **Campaign** – beta – Campaign
- **CampaignResponse** – beta – Campaign Response
- **CampaignTemplate** – beta – Campaign Template
- **CashRefund** – beta – Cash Refund
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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

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

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

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

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

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

- DownloadItem – beta – Download Item

- GlobalTransaction – beta – Global Transaction
All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- InboundShipment – beta – Inbound Shipment
- IntercompanyJournalEntry – beta – Intercompany Journal Entry
- InventoryAdjustment – beta – Inventory Adjustment
- InventoryCostRevaluation – beta – Inventory Cost Revaluation
- InventoryCount – beta – Inventory Count
- InventoryItem – beta – Inventory Item
- InventoryNumber – beta – Inventory 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

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

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

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.
All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- KitItem – beta – Kit Item

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

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

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- ManufacturingCostTemplate – beta – Manufacturing Cost Template
- ManufacturingOperationTask – beta – Manufacturing Operation Task
- ManufacturingRouting – beta – Manufacturing Routing
- MapReduceScript – beta – Map Reduce Script
- MarkupItem – beta – Markup Item
- MerchandiseHierarchyLevel – beta – Merchandise Hierarchy Level
- MerchandiseHierarchyNode – beta – Merchandise Hierarchy Node
- MerchandiseHierarchyVersion – beta – Merchandise Hierarchy Version
- Message – beta – Message
- MfgPlannedTime – beta – Mfg Planned Time

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- Nexus – beta – Nexus
- NonInventoryItem – beta – Non-Inventory Item
- Note – beta – Note
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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

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

- 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
- Portlet – beta – Portlet
- PriceBook – beta – Price Book
- PriceLevel – beta – Price Level
- PricePlan – beta – 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

R

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- ReceiveInboundShipment – beta – Receive Inbound Shipment
- ResourceAllocation – beta – Resource Allocation
- Restlet – beta – Restlet
- RevenueArrangement – beta – Revenue Arrangement
- RevenueCommitment – beta – Revenue Commitment
- RevenueCommitmentReversal – beta – Revenue Commitment Reversal
- RevenuePlan – beta – Revenue Plan
- RevRecSchedule – beta – Rev Rec Schedule
- RevRecTemplate – beta – Rev Rec Template

S

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- SalesOrder – beta – Sales Order
- SalesRole – beta – Sales Role
- SalesTaxItem – beta – Sales Tax Item
- ScheduledScript – beta – Scheduled Script
- ScheduledScriptInstance – beta – Scheduled Script Instance
- ServiceItem – beta – Service Item
- ShipItem – beta – Ship Item
- Solution – beta – Solution
- StatisticalJournalEntry – beta – Statistical Journal Entry
- StorePickupFulfillment – beta – Store Pickup Fulfillment
- Subscription – beta – Subscription
- SubscriptionChangeOrder – beta – Subscription Change Order
- SubscriptionLine – beta – Subscription Line
- SubscriptionPlan – beta – Subscription Plan
- Subsidiary – beta – Subsidiary
- SubtotalItem – beta – Subtotal Item
- Suitelet – beta – Suitelet
- SupplyChainSnapshot – beta – Supply Chain Snapshot
- SupplyChainSnapshotSimulation – beta – Supply Chain Snapshot Simulation
- SupportCase – beta – Supply Case

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- 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

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- UnitsType – beta – Units Type
- UserEventScript – beta – User Event Script

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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- Vendor – beta – Vendor
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.

All records available in SuiteScript are available in REST web services, too. All records in REST web services are considered beta until indicated otherwise.

- **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** – beta – Website
- **WorkflowActionScript** – beta – Workflow Action Script
- **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

The following ERP and SuiteBilling records are supported in REST web services:

ERP Records
- Subsidiary

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

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

```
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.

```json
{
  "subscriptionplan": {
    "id": "{{existing_subscription_plan_id}}"
  }
}
```

```json
{
  "subscriptionplan": {
    "id": "{{existing_subscription_plan_id}}"
  },
  "currency": {
    "id": "1"
  },
  "name": "My Price Book {{timestamp}}",
  "priceinterval": {
    "items": [
      {
        "priceplan": {
          "id": "{{last_priceplan}}"
        },
        "frequency": "MONTHLY",
        "repeatevery": "1",
        "startoffsetvalue": 1
      }
    ]
  }
}
```
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

```json
{"currency": {"id": "1"},
"priceplantype": "2",
"pricetiers": {
"items": []
}
```
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",
    ...
}
```
Create with Different Pricing Values

```json
{
  "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": 820},
        "frequency": "ANNUALLY"
      }
    ]
  }
}
```

Create a New Price Interval

```json
{
  "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": 820},
        "frequency": "ANNUALLY"
      }
    ]
  }
}
```
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,
                "includeinrenewal": true,
                "revenueoption": "OVER_TERM"
            },
            {
                "linenumber": 2,
                "proratestartdate": false,
                "proratedenddate": false,
                "includeinrenewal": false
            }
        ]
    }
}
```

Create with an Add-On Item

```
{
    "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": 4,
                "includeinrenewal": true,
                "item": 5,
                "subscriptionlinetype": 2,
                "billingMode": "IN_ADVANCE",
                "proratestartdate": false,
                "proratedenddate": false
            }
        ]
    }
}
```
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,
   "isincluded": false
}

"priceinterval": {
   "items": {
      "subscriptionplanlinenumber": 4,
      "linenumber": 4,
      "quantity": 3,
      "repeatEvery": 2,
      "pricePlan": {"id": 1217},
      "frequency": "ANNUALLY"
   }
}
}
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": [
            {"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
The following subsidiary subrecord elements are exposed to REST web services:

- (email) - return email address
- (fax) - fax

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.

```
{GET /services/rest/record/v1/subsidiary/<id>}

{GET /services/rest/record/v1/subsidiary?q=fullName CONTAIN "parent"}
```
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:
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

\[\textbf{Note:} \text{ At present time, there is no way to enforce this behavior through NetSuite, so you can make the lines optional and let the sales person enforce this rule.}\]

- 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 \textbf{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"
      }
    ]
  }
}
```
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
      }
    ]
  }
}
```
Use Case: Manage Your Subscription Catalog

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": {
    "items": [
      { "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": {
    "items": [
      { "fromVal": 0,
        "pricingOption": { "id": "-102" },
```

SuiteTalk REST Web Services Records Guide
Price Plan 5

```json
{
    "currency": { "id": "1" },
    "pricePlanType": "2",
    "priceTiers": {
        "items": [
            {
                "fromVal": 0,
                "pricingOption": { "id": "-102" },
                "value": 20.00
            }
        ]
    }
}
```

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
            }
        ]
    }
}
```
Use Case: Manage Your Subscription Catalog

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
{
  "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",
        "startOffsetValue": 1,
        "prorateBy": "DAY"
      },
      {
        "subscriptionPlanLineNumber": 4,
        "pricePlan": { "id": "13" },
        "frequency": "MONTHLY",
        "repeatEvery": "1",
        "startOffsetUnit": "YEAR",
        "startOffsetValue": 1,
        "prorateBy": "DAY"
      }
    ]
  }
}
```
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.suiteetalk.api.netsuite.com/services/rest/record/v1/subscriptionplan/1

Returns this response body:

```
{
    "links": [
    {
        "rel": "self",
        "href": "http://demo123.suiteetalk.api.netsuite.com/services/rest/record/v1/subscriptionplan/1"
    }
    ],
    "autoRenewal": false,
    "createdDate": "2020-1-1T00:00:00Z",
    "customForm": "-950",
    "defaultRenewalTerm": {
        "links": [],
        "id": "1",
        "refName": "Three Years"
    },
    "id": "1",
    "includeChildren": false,
    "incomeAccount": {
        "links": [
        {
            "rel": "self",
            "href": "http://demo123.suiteetalk.api.netsuite.com/services/rest/record/v1/account/1"
        }
        ],
        "id": "1",
        "refName": "Sales"
    },
    "initialTerm": {
        "links": [],
        "id": "1",
        "refName": "Three Years"
    },
    "isInactive": false,
    "itemId": "DEFEND Package",
    "lastModifiedDate": "2020-1-1T00:00:00Z",
    "member": {
        "links": [
        {
            "rel": "self",
            "href": "http://demo123.suiteetalk.api.netsuite.com/services/rest/record/v1/subscriptionplan/1/member"
        }
        ]
    }
}```
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

```json
{
  "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

```json
{
  "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**

```
{
  "subscriptionLine": {
    "items": [
      {
        "lineNumber": 4,
        "isIncluded": true
      }
    ],
  }
  "priceInterval": {
    "items": [
      {
        "subscriptionPlanLineNumber": 1,
        "discount": 50.0
      },
      {
        "subscriptionPlanLineNumber": 3,
        "quantity": 25
      }
    ]
  }
}
```

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:
Use Case: Manage Your Subscription Sales

- 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**

```json
{
    "subscriptionLine": {
        "items": [
            {
                "lineNumber": 7,
                "item": { "id": "7" },
                "subscriptionLineType": "2",
                "billingMode": "IN_ADVANCE",
                "renewalOption": "ALWAYS",
                "prorateStartDate": true,
                "prorateEndDate": true
            }
        ],
        "priceInterval": {
            "items": [
                {
                    "subscriptionPlanLineNumber": 7,
                    "pricePlan": { "id": "19" },
                    "frequency": "MONTHLY",
                    "repeatEvery": "1",
                    "startOffsetUnit": "MONTH",
                    "startOffsetValue": 1,
                    "prorateBy": "DAY"
                }
            ]
        }
    }
}
```