Babak Rezvani

Software Documentation Writing Samples

Contents

Introduction to Writing Samples

Reviewer

The following topics are truncated samples of my writings. The samples are divided into the following categories:

- *API*
- *Conceptual*
- *Procedural*

***Note**: To prevent proprietary issues, the names of the applications and guides have been replaced with generic names.*

API Writing Sample

Reviewer

The following topics are samples of four chapters and two related topics taken from the Integration Interface Guide (A.K.A. API document).

Target audience

Developer who wants to develop applications which interact with the application.

Background and context

I began working on the Integration Interface Guide in 2014 after the writer assigned to maintaining the guide retired.

At that time, the guide only covered one type of web service, the Simple Object Access Protocol (SOAP) and a handful of supporting information.

For a while, I basically maintained the contents that were affected by the product updates (i.e., mostly basic operations updates or adding new ones).

In the first half of 2016, I reorganized the contents and updated the titles (to have a more descriptive content), in addition to incorporating changes or enhancements.

In the second half of 2016, the application was deployed for the web, in addition to the on-premises deployment, and RESTful web services were built to enable the application to work on the web because REST allowed the usage of different data formats (HTML, XML, JSON, plain text, etc.), unlike SOAP that used XML format only.

Since then I have been responsible for adding REST web services to the API document.

Introduction

The project management application supports integration with multiple applications.

The interface to integrate the application and an external system is based on:

- Simple Object Access Protocol (SOAP) and
- Representational State Transfer (REST) web services.

This guide describes how to use the web services.

> **Note**: We recommend that the new integrations to use RESTful web services, only.

Within our documentation, some content might be specific for cloud deployments while other content is relevant for on-premises deployments. Any content that applies to only one of these deployments is labeled accordingly.

Prerequisites

Integration with the application primarily depends on:

- The applications selected to integrate data with the application
- The availability of the selected applications, as a cloud service or on-premises
- The setup and configuration of integration:
  - within application, or
  - external to the application.

Methods

The API supports the following HTTP request methods:

GET	Returns objects or values.
DELETE	Deletes objects or values.
POST	Creates objects or values.
UT	Updates objects or values.

After receiving a request, the application invokes the business and security logic to service the received request by providing the appropriate response.

Simple Object Access Protocol (SOAP) Web Services

The SOAP web services enables you to exchange structured data between the environments. The structure of the exchanged data is specified by an XML Schema.

> **Note**: Use SOAP with Attachments API for Java™ (SAAJ) as the Application Programming Interface (API) when you decide to write SOAP messaging applications, directly.

You can use the following methods for the SOAP web services:

- Resource Manager Methods
- Space Manager Methods
- Planning Manager Methods
- User Administration Methods
- Schedule of Values (SOV) Methods
- Exchange Rates Methods
- Configurable Manager Methods
- Shell Methods
- Schedule Manager Methods
- Asset Manager Methods
- User Defined Reports (UDR) Methods
- Cost Sheet Methods
- Project Methods
- Business Process (BP) Methods
- CBS Code Methods

Each method is described below.

Resource Manager Methods

Create or update Resource

Description

This method creates or updates the resources in the **Resource Manager**.

Support

This method supports creating or updating the resources at the company level.

This method does not support creating or updating the resources at the shell or project level.

Installation

Application Service Provider (ASP)

Self-host

Input Request (Prototype)

```
public XMLObject createUpdateResource
   shortname (String)
   authcode (String)
   resourceXML (String)
```

Parameters

shortname

Identifier of the company name or company short name.

authcode

The authentication key for the company (in text string).

resourceXML

The information of the resources that will be created or updated.

Return Value

See "Appendix A: Return Values" in this guide for the description of all of the return values.

Sample

```
createUpdateResource("acme", "acme_authcode", XML)
```

Additional Information

The "_resource" tag can be repeated to create multiple resources.

END

Representational State Transfer (REST) Web Services V1

REST web services are designed to be used on the web. REST uses the Hypertext Transfer Protocol (HTTP) to communicate data on the web, and to attain the resources data by way of accessing the Uniform Resource Identifiers (URIs), or web links.

> **Note**: If the endpoint URL has a project number and the project number contains special characters (such as / \ : * ? " < > |), then you need to change those special characters with URL escape characters.

Use the following principles for creating your RESTful application:

- Uniform interface
- Self-descriptive messages
- Self-descriptive messages
- URIs

The following topics contain pertinent details for creating RESTful application:

- Document Manager
- Business Processes
- Shell Manager
- Level
- Space
- Cost
- Cash Flow
- Schedule Sheet
- Exchange Rates
- Data Structure Setup
- Partner Company
- User
- User Defined Report (UDR)
- Get Templates List
- Data
- Data Format
- Data Transfer
- Event Driven Notification
- Funding
- Non-Workflow BP Permissions

Cost

Get Column Data

GET

`/ws/rest/service/v1/cost/columndata/{project_number}`

Purpose

To get the column data for all of the rows, or the line items, of the cost sheet (for the specified column name).

Input

All parameters must be URL encoded.

Path parameter

`project_number` (Required)

Specify the project number to get the column data of the project cost sheet.

Query parameter

`columnname` (Required)

Specify the column name of the project cost sheet.

Body

Use JavaScript Object Notation (JSON)

Output

Body

JSON

The JSON object must contain: 'status', 'data', 'message'

Get Column Data Output (JSON)

```
{
  "data": [
    {
      "wbs_code": "Cost Code 1",
      "short_description": "Direct Entry Line Item",
      "amount": "1000.0",
      "spends_category": "",
      "quantity": "10.0",
      "work_package": "",
      "unit_of_measure": "each",
      "unit_cost": "100.0",
      "long_description": ""
    },
    {
      "wbs_code": "Cost Code 2",
      "short_description": "Direct Entry Line Item",
      "amount": "100.0",
      "spends_category": "",
      "quantity": "10.0",
      "work_package": "",
      "unit_of_measure": "",
      "unit_cost": "10.0",
      "long_description": ""
    }
  ],
  "message": [
    "success"
  ],
  "status": 200
}
```

Status Codes

Success: 1 > 200

END

Conceptual Writing Sample

Reviewer

The following topics are two samples explaining two dissimilar subjects conceptually.

Target audience

All application users (users, aAdmins, and form designers) should read this guide.

Background and context

The Managers User Guide, part of application Help, explains how to work with the application modules and managers.

API

Schedule Manager Methods

Create Schedule Sheet Activities

Additional Information About Creating Schedule Sheet Activities

You must perform the required data mapping process on the target schedule sheet. Ensure that you:

- map the schedule sheet columns to Primavera XML elements, and
- set the appropriate XML import options.

> **Note**: The system copies only the elements that are mapped.

Use the **Create Schedule Sheet Activities** method to create new activities. This method is not available for updating schedule sheet activities. For updating the schedule sheet activities, see the "Update Schedule Sheet Activities from Application" topic in this guide.

If you invoke (execute or call) the **Create Schedule Sheet Activities** method in an empty schedule sheet, then the system inserts activities into the empty schedule sheet.

If you invoke the **Create Schedule Sheet Activities** method in a non-empty schedule sheet, then the system overwrites all of the existing activities (delete and recreate) in the non-empty schedule sheet, if the function is allowed in the data mapping setup.

If you invoke the **Create Schedule Sheet Activities** method in a non-empty schedule sheet, and the data mapping setup does not allow overwrite, then the system will return an error, in the response.

Use the **Create Schedule Sheet Activities** method for projects of cost-code type Cost Breakdown Structure (CBS).

When an activity is created by way of the **Create Schedule Sheet Activities** method, the system:

- assign a Globally Unique Identifier (GUID) to the activity and
- uses a default data mapping.

END

Project

About Business Process Forms

Business processes (BPs) can be designed with or without workflows, so there are two types of BP forms in the application:

- Workflow
- Non-workflow

Workflow BP Form

A workflow specifies how a BP should proceed, from start to finish. It also illustrates each step in the BP and displays the behavior of each step.

A workflow also includes the movement of documents around the organization for purposes including sign-off, evaluation, or performing activities in a process.

A workflow BP has three sections:

- Upper Form
- Workflow
- Content

The following explains each section in detail.

Upper Form

The Upper Form section of a Workflow BP form contains the basic information the form is managing, such as the name of a record and its description, who created the record and when, and so on. It also contains general data entry, informational, summary, or reference fields.

Workflow (Task Details)

The Workflow or Task Details section of a Workflow BP form shows the details of the step the form is currently on, including the last person who took action on the form, the status of the step, the due date for the task, and any notes that have been included at this step. Depending on how the workflow has been set up, the next assignee(s) for the task may be pre-determined (as in the example above), or the user may need to specify the next assignee or send a copy of the form to another user.

Content (Lower Form)

The Content or Lower Form section of a Workflow BP form shows information that has already been attached to the form, such as a line item, a document, or response comments for an RFI. The information displayed in this part of the form is entered by way of a Detail Form that the user completed when the user accepted the task.

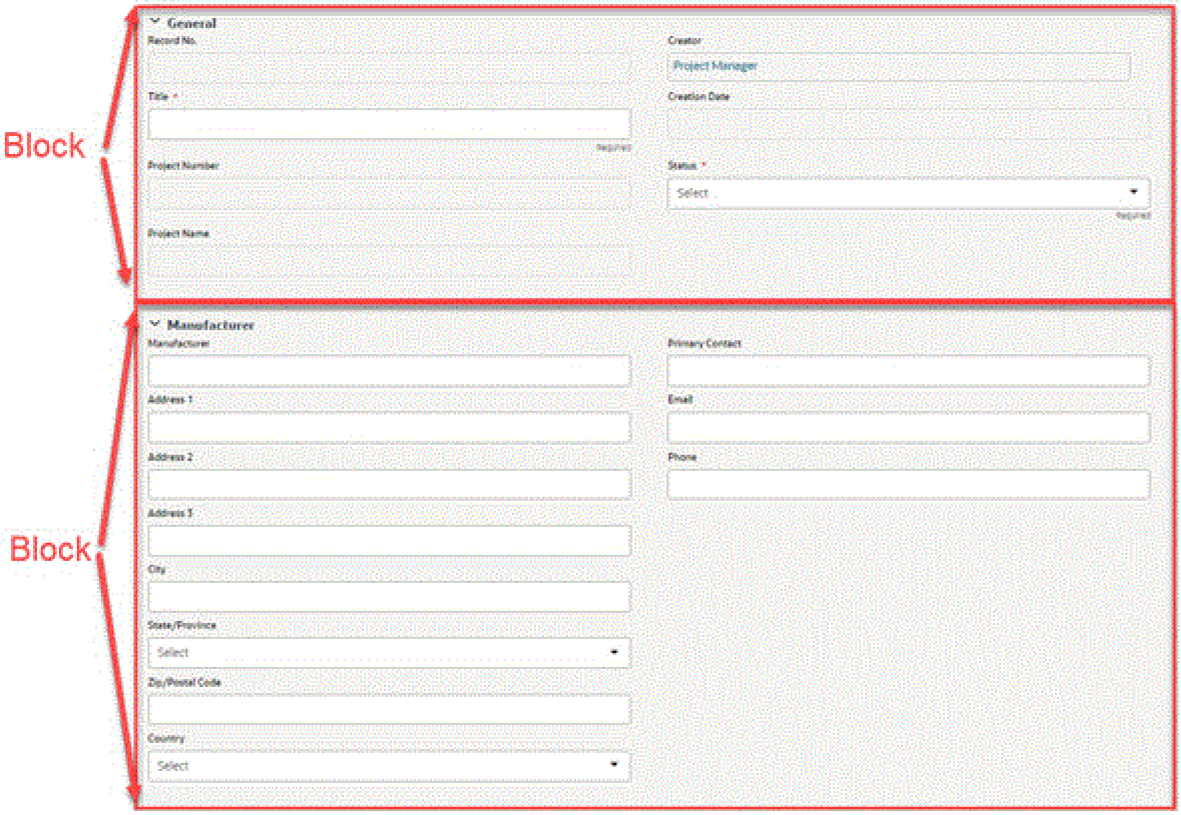
Non-workflow BP Form

Most BPs will include a workflow or workflows; however, some BPs have a single purpose of storing data. These workflows are often data entry forms designed to enter information directly into the system (for example, vendor lists). These business processes are called non-workflow BPs.

In terms of the form construction, the non-workflow BP form is similar to the workflow BP; however, the non-workflow BP form allows you to toggle between read-only and edit modes.

What Goes on a Form

Forms are composed of blocks and fields. Each block can contain one or more fields which you can fill with data elements.



Data Elements (DEs)

A DE can be:

- A text box, where the user types in information.
- A drop-down (pull-down) menu of choices (also known as a picker) such as dates or names.
- Radio buttons, where the user must select one of the options presented.
- A check box, where the user has the option of choosing something or not.

Refer to the *Data Reference Guide* for a list of these elements, along with a description of what they do and where to use them.

You can also add new elements as your company requires them. For instructions on creating new data elements, see the "Creating a Data Element" topic in this guide.

Pickers

A picker is a field that displays database records returned by auto-population, reverse auto-population, or pre-configured SQL query from an integrated database.

The application contains a substantial number of pickers. These pickers are important elements for the forms and a basic understanding of them will help you know where to include them on your forms.

Pickers appear as selection lists on BPs and in several other application components.

Pickers allow the user to choose elements such as dates, blanket purchase orders (POs), other users, companies, line items, funds, or currencies.

Refer to the *Data Reference Guide* for a list of pickers, along with a description of what they do and where to use them.

END

Procedural Writing Sample

Reviewer

The following topics are two samples explaining two dissimilar subjects procedurally.

Target audience

All users (users, admins, and form designers) should read this guide.

Background and context

The Earned Value Management User Guide, part of application Help, explains how to calculate the earned value and related measures for your project.

This was a brand new guide, and I wrote this guide without utilizing any previous work, and by way of working with the application, reviewing design demos, and having one-on-one meetings with the SME.

Activity Manager

Manual Activity Sheet

The manual activity sheet is mainly used for:

- Projects (such as building owners projects) that do not use an external schedule application integration
- Project schedules (simple project schedules)

> **Note**: The maximum duration of an activity in a **Manual Activity Sheet** is five years (rounded to 20000 working hours).

You can create an activity sheet, manually, in the **Activity Sheets** log (**User** mode), and if you have the pertinent permissions, the **Activity Sheets** log displays the following options in the log window, in addition to the other options:

- Receive and send the scheduling and resource data into an activity sheet from P6 or Primavera Cloud
- Create activity sheets both manually and through synchronization with P6 or Primavera Cloud

To be able to create and work with manual activity sheets review the following information about the options that are available within the **Activity Sheets** log, first:

Create

If you have the **Create Manual Activity Sheets** permission, or the **Full Access** permission, you will see the **Create** option in the log window.

If only the standard activity attribute form is defined but not manual activity attribute form, then the **Create** option in the **Activity Sheets** log is not displayed.

To create an activity sheet, you must use the **Create** option.

Get Data (⇥)

If you have permission to use **Get Data** only, then you will see the **Get Data** option in the log window.

This option enables you to get data into the System Activity Sheet.

The get data synchronization that created the activity sheet also populates the activity sheet with the:

- Scheduling data (activities along with assignments and spread data) from the P6 projects mapped to the current shell in the **Integration** tabs (**Gateway Integration** for P6 and **Primavera Cloud Integration** for Open Platform Communications (OPC)).
- Role and resource rates data from the **Master Rate Sheet** for the mapped projects.

The subsequent updates of **CBS Code**, **Role Rate**, and **Resource Rate** in P6 will not trigger the update of the:

- **CBS Code** (in the **Activity Sheet**)
- **Role Rate** and **Resource Rate** (in the **Master Rate Sheet**). This is to prevent data in the application from being overwritten by updates in P6.

Send Data (⇥)

This option enables you to send data from the System Activity Sheet.

> **Note**: The **Get Data** and **Send Data** options are used to create a **System Activity Sheet**.

To create a manual activity sheet

1) In the **Activity Sheets** log, click **Create** to open the **Create Activity Sheet** window.
2) Proceed to name the manually created activity sheet, determine the status, select a calendar, select the source for the project schedule start date, select the schedule start date, data date, and the activity sequence number.

   The **Time Zone** and **Schedule Type** selected in the properties will be saved and cannot be reverted. The **Time Zone** defaults to the time zone indicated in the **User Preference**, and the **Data Date** defaults to the Project Schedule Start Date.
3) When you are done, click **Create**, and click **Yes** when the confirmation message appears. The **Create Activity Sheet** window opens.
4) Proceed to enter values in the following fields, on the **Create Activity Sheet** window and when you are finished click **Save**.

Field	Description
Name	Use this required field to specify the name of the activity sheet. The name specified should be unique. **Note**: Do not use "System Activity Sheet" as the name. The field supports names of 150 characters or less.
Description	Use this long-description field to describe the activity sheet. This field supports alphanumeric characters, as well as all special characters, and can be up to 400 characters in length.
Status	Use this field to make an activity sheet active or inactive. The default status is Active. The activity sheet can later be set as active or inactive through the properties screen. Only the active sheet data will be rolled up to other cost modules like Cost Sheet, EVM Sheet, Cash Flow, and so on.
Calendar	This field lets you select any calendar for the project schedule. You cannot deselect the calendar from the drop-down, but you can change the selected calendar. This field has the following options to select from: ▸ **Standard** (24x7 Calendar or Company Calendar) ▸ **Project** / **Shell** (Project/Shell Calendar) ▸ **Custom** (Custom Calendar) The Project/Shell Calendar is the default value. The drop-down field lists all of the calendars that have been defined at the **Custom Calendar** node, in the shell, project, and company calendars (as defined in the **Standards & Libraries**).
Select Project Schedule Start Date Source	This field lets you select a manual start date, or the start date in the shell properties, for your project (project schedule start date). **Manual** **From Shell Attribute** The value in this field is identified as the start date for the project schedule, as defined in the activity sheet. The project schedule start date can be entered manually by selecting the **Manual** option or by using the shell attribute: **uuu_project_start_date**. This is a required field in the activity sheet properties. If it is selected as manual, then you must provide a start date in order to be able to save the activity sheet properties. The second option, **From Shell Attribute**, will populate the start date on the activity sheet by using the targeted project start date (**uuu_project_start_date**) on the shell or project properties which in turn can be received from the project creation business process. Any change in the project start date in shell properties will automatically update the activity sheet start date. Synchronous update to the activity sheet takes place as long as there is no activity which is either in-progress or has been completed.

Field	Description
Schedule Start Date	If you select Manual in the previous field, this field lets you enter a specific start date for the project schedule.
Time Zone	The field defaults to the User Preference time zone. This time zone is used when saving the activities date and time within the sheet. The users or groups who have view access to this sheet will see the activity dates and times based on the specified time zone.
Technique for computing Estimate to Complete (ETC)	This field is available when you are creating an activity sheet, or when an ETC user-value must be added for the user to select the ETC technique, as explained below. By default, the ETC technique for the activity sheet is applicable for all of the Work Breakdown Structure (WBS) codes and activities, unless there is a different ETC technique for the WBS or activity. The ETC drop-down field will have the following values: ETC = remaining cost for the activity, the default setting ETC = [PF*(Budget at Completion- Earned Value Cost)] PF = 1 PF = 1/CPI PF = 1/(CPI * SPI) PF = user defined value After creating the activity sheet manually by defining the properties, the system creates a default activity sheet. You can then update the activities. (If an activity sheet was created by integration, it contains activities that you can subsequently update.) See the "Manual Activity Sheet Tabs" topic in this guide for details on activity properties.
Continued ...	

END

Installing Application on Linux (On-Premises)

Complete the following steps to install and configure the application for a first time installation.

Each step corresponds to a section in this guide.

Note: Before you begin, create an installation account that has full administration privileges for the server. This account is needed for maintenance and upgrades.

1) Download application
2) Configure the database server

   You must complete this step before configuring the application because the system uses the information in this step during database configuration in the application Configurator.
3) Configure WebLogic Server cluster on Linux
4) (Optional) Install and configure the Reports Server
5) (Optional) Install WebCenter Content
6) Install AutoVue Server
7) Configure the application by using the Configurator
8) Configure the Web Server
9) Deploy the application in WebLogic
10) Launch the application and proceed to install the modules

END

Configure WebLogic Server Cluster on Linux

To create a WebLogic domain:

1) Run the WebLogic **Configuration Wizard**.
2) In the **Welcome** window:
   a. Select **Create a new WebLogic** domain.
   b. Click **Next**.
3) In the **Select Domain Source** window, click **Next** to accept the default selections.
4) In the **Specify Domain Name and Location**:
   a. Enter a domain name for the new domain to be created.
   b. Enter the location of the new domain on the server.
   c. Click **Next**.
5) In the **Configure Administrator User Name and Password** window:
   a. Enter values for the administrator **User Name** and **Password** fields. The user name will be used to login to the WebLogic console.
   b. Click **Next**.
6) In the **Configure Server Start Mode and JDK** window:
   a. In the left pane, select **Production Mode**.
   b. In the right pane, select the Java™ Development Kit (JDK™) that you installed earlier.
   c. Click **Next**.
7) In the **Select Optional Configuration** window:
   a. Select the **Administration Server** option.
   b. Click **Next**.
8) (Optional) In the **Configure the Administration Server** window:
   a. Select the Secure Sockets Layer (SSL) enabled option. You can proceed to set the SSL port if you are enabling SSL communication.

      Note: We recommend that you always use SSL in the application Production environment for secure communications.

   b. Click **Next**.
9) In the **Configuration Summary** window, click **Create**.
10) Click **Done** after the domain is created.
11) When prompted, enter the **Administrator User Name** and **Password** that you entered above.

END

Editing SetEnv.sh File on Linux

Ensure that the following variables are saved in the **setenv.sh** file:

- Set the `domain_home` variable to specify the path of the domain home folder that will be used by the application.
- Set the `admin url` variable to specify host name and port number used by the application domain.
- Set the `java_home` variable to the JDK installed directory.
- Set the `JAVAFX_HOME` variable to the JAVAFX installed directory, if you are using JDK 11 and later versions.
- Set the `USER_MEM_ARGS` variable specify the JVM maximum memory setting.

END