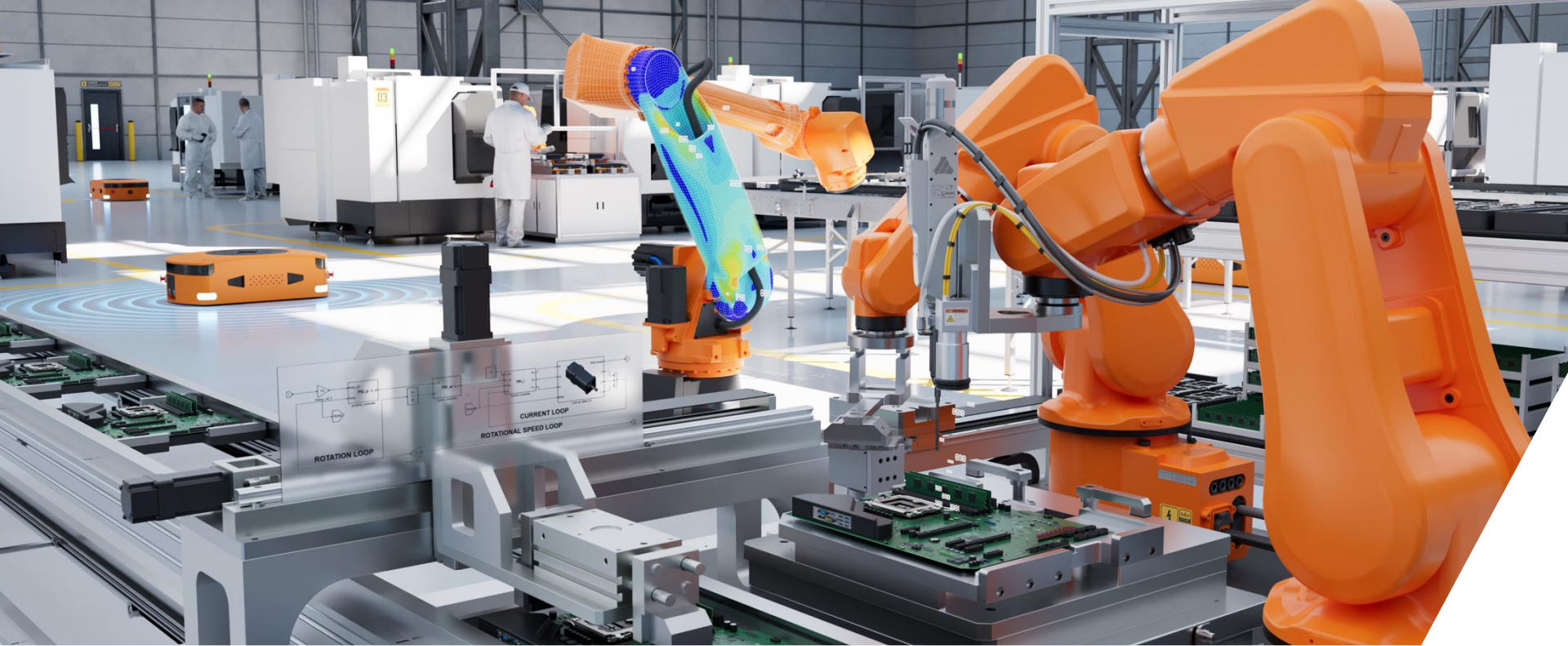




Effective APIs for SAS Language Applications

Randy Betancourt, Senior Director, Altair

PHUSE: EU-Connect 2024

Safe Harbor Statement

This presentation and the accompanying oral commentary may contain “forward-looking” statements that are based on our beliefs and assumptions and on information available to us as of the date of this presentation. All statements other than statements of historical facts contained in this presentation, including statements regarding our future results of operations and financial position, customer lifetime value, strategy and plans, market size and opportunity, competitive position, industry environment, potential growth opportunities and our expectations for future operations, are forward-looking statements. The words “believe,” “may,” “might,” “objective,” “ongoing,” “will,” “estimate,” “continue,” “anticipate,” “design,” “intend,” “expect,” “could,” “plan,” “potential,” “predict,” “project,” “seek,” “should,” “would” or the negative version of these words and similar expressions are intended to identify forward-looking statements. This presentation also contains non-GAAP financial measures. We have provided a reconciliation of such non-GAAP financial measures to the most directly comparable measures prepared in accordance with U.S. GAAP in the Appendix to this presentation.

We may not actually achieve the plans, intentions or expectations disclosed in our forward-looking statements. Forward-looking statements involve known and unknown risks, uncertainties, assumptions and other factors that may cause our actual results, performance, achievements or expectations to be materially different from any future results, performance, achievements or expectations expressed or implied by the forward-looking statements. Except as required by law, we assume no obligation to update these forward-looking statements publicly, or to update the reasons why actual results could differ materially from those anticipated in the forward-looking statements, even if new information becomes available in the future.

Effective APIs for SAS Language Applications--Agenda

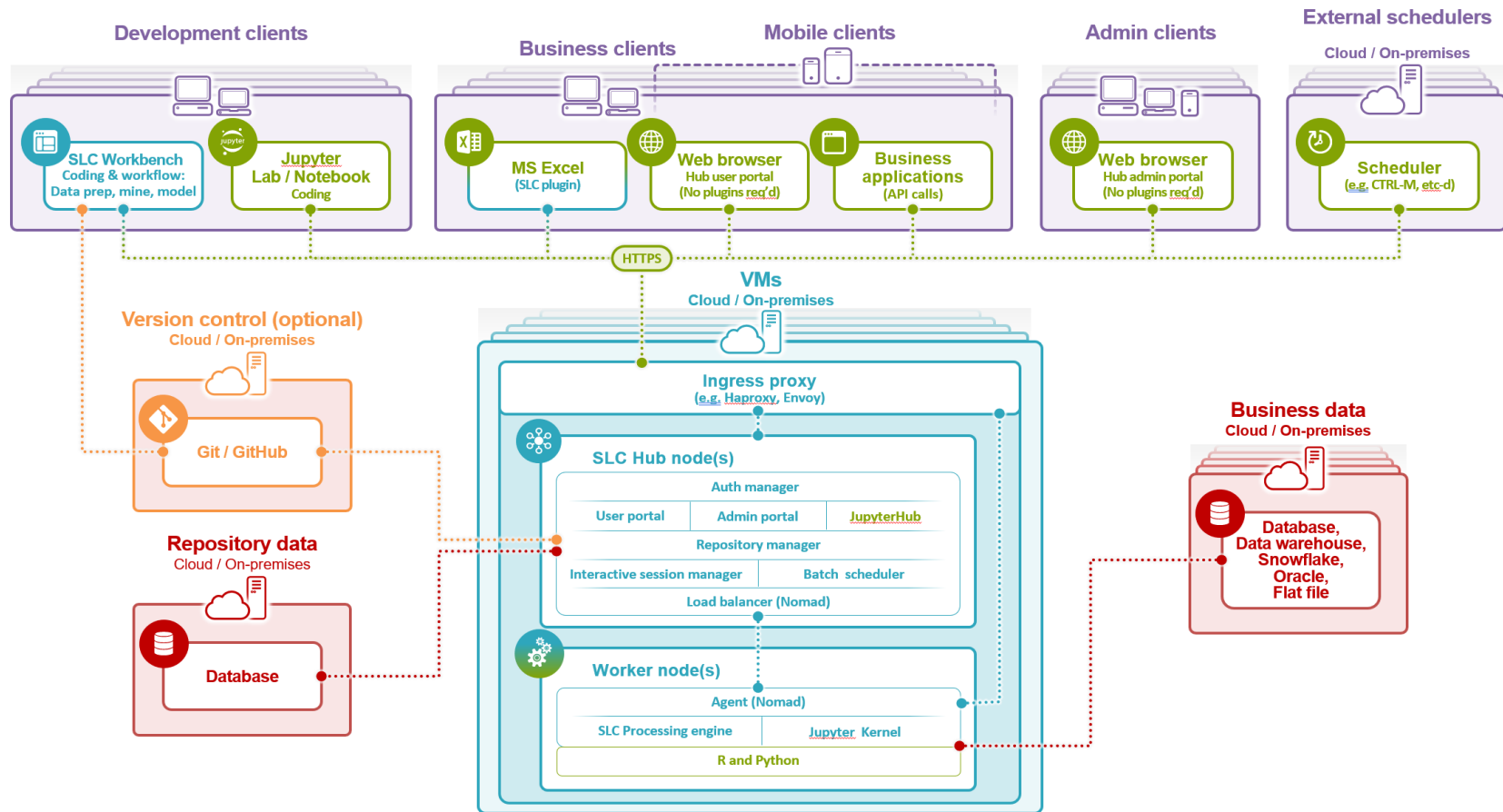
REST APIs for Abstraction and Code Reuse

- REST API Basics
 - Use cases for REST APIs
 - REST API Design Considerations
-
- SLC Hub Software Architecture—Enablement
 - Authentication & Authorization Considerations
 - SAS Language REST API Design (walk-through)
 - Client Consumption of SAS Language REST APIs

REST API Use Cases

- REST APIs encourage code reuse through abstraction
- Servers hosting REST APIs have “listener” processes—implying some administrative overhead
- Clients access resources as URLs & servers return output as JSON, XML, etc.
- REST APIs are stateless—requiring good design to permit scalability
- Authentication for client identify and authorization for resource access must be defined
- SAS AUTOCALL Macros are ideal candidates for exposing as REST APIs

SLC HUB ARCHITECTURE



HUB PORTAL ADMINISTRATION

Altair SmartWorks Hub™

Home

?

Namespace1

Randy Betancourt

Welcome to the Altair SmartWorks Hub Portal



Deployment Services

Manage batch execution pipelines and deploy packages of programs that are invoked via an HTTP request, as batch jobs, or from the Altair SmartWorks Hub plugin f...

Launch Deployment Services



Enterprise

Tools to assist large scale deployments of Altair SmartWorks Hub

Launch Enterprise



Hub Administration

Administer Users and groups; perform system configuration and monitoring

Launch Hub Administration



Invocation

Execute programs deployed via Altair SmartWorks Hub Deployment Services

Launch Invocation

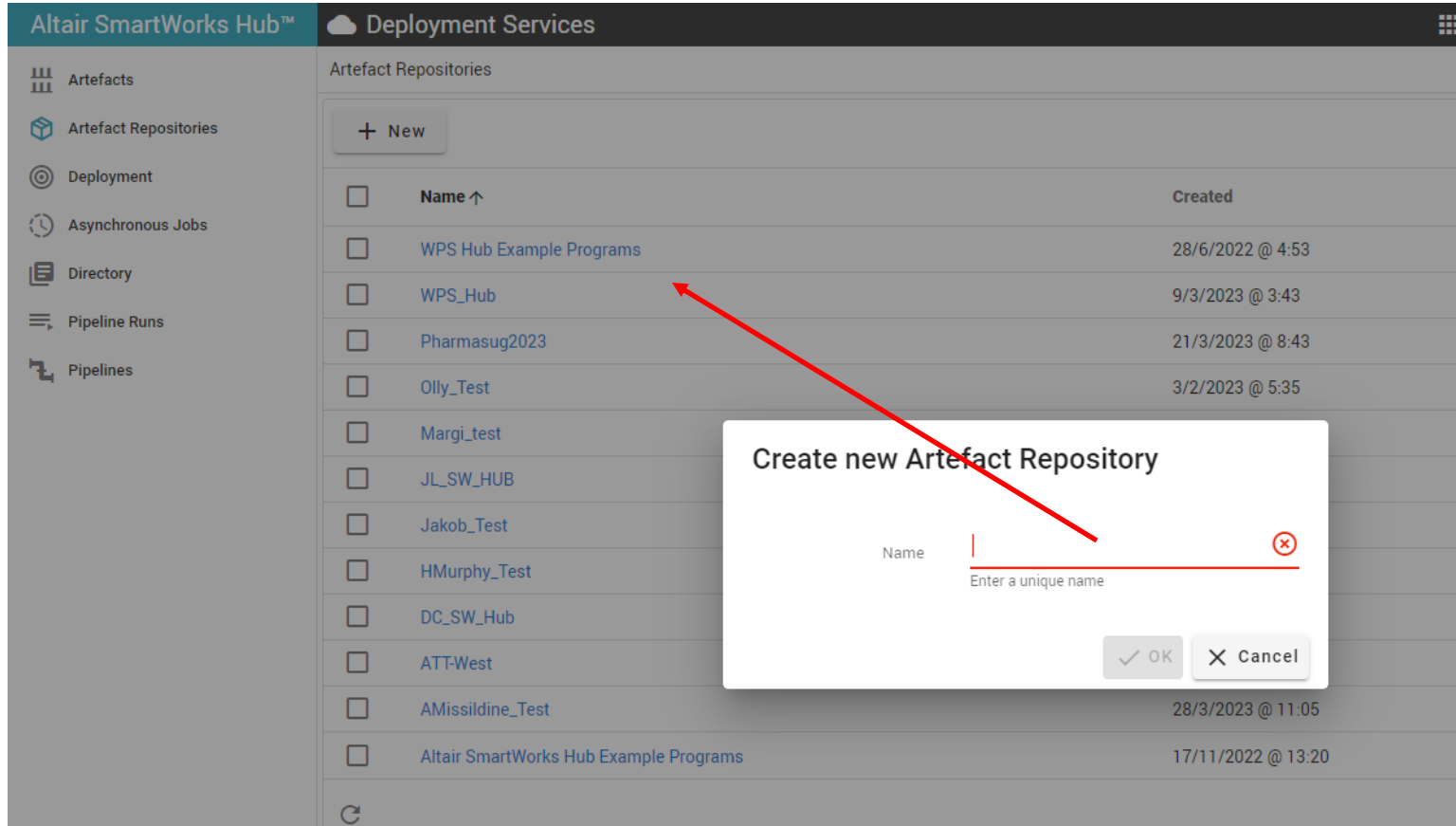
↑

API Design Tasks

↑

API Execution/Control Tasks

Define a Repository



The screenshot displays the Altair SmartWorks Hub interface. On the left is a sidebar with navigation options: Artefacts, Artefact Repositories, Deployment, Asynchronous Jobs, Directory, Pipeline Runs, and Pipelines. The main header is 'Altair SmartWorks Hub™' and the main title is 'Deployment Services'. Below the title is a sub-header 'Artefact Repositories' and a '+ New' button. A table lists existing repositories with columns for Name and Created. A modal dialog titled 'Create new Artefact Repository' is open in the foreground, featuring a 'Name' input field with a red error indicator and a 'Create' button. A red arrow points from the 'Name' input field to the 'WPS_Hub' entry in the background table.

Name ↑	Created
WPS Hub Example Programs	28/6/2022 @ 4:53
WPS_Hub	9/3/2023 @ 3:43
Pharmasug2023	21/3/2023 @ 8:43
Olly_Test	3/2/2023 @ 5:35
Margi_test	
JL_SW_HUB	
Jakob_Test	
HMurphy_Test	
DC_SW_Hub	
ATT-West	
AMissildine_Test	28/3/2023 @ 11:05
Altair SmartWorks Hub Example Programs	17/11/2022 @ 13:20

Create new Artefact Repository

Name

Enter a unique name

OK Cancel

Create a parameter-drive SAS Program

```
/* generate_dates.sas */  
data dates;  
  
do date = "&start_date" to "31Dec2023"d;  
    output;  
end;  
  
format date YYMMDDd10.;  
run;
```

DESKTOP: Configure API Parameters for HUB

© Altair Engineering Inc. Proprietary and Confidential. All rights reserved.

Workspace2 - Pharmsug2023/generate_dates.sas - Altair Analytics Workbench

File Edit Navigate Search Project Run Altair Window Help

Log for Loc... write_read... regression... ODS_mod_tem... proc_reg_ex... *generate_d...

Hub Configuration

General Information

Label: RunGenerateDates

Program path: RunGenerateDates

Parameters

Parameter style: Macro variables

Type name filter text

Type	Name
Date	start_date
String	
Password	
Choice	
Integer	
Float	
Boolean	
Stream	
Date	
Time	
Date & time	

Add Remove Up Down

Parameter Details

Name: start_date

Label:

Prompt:

☐ Hidden

☒ Required

Type: Date

☒ Provide default date

Default date: 30/12/2023

☐ Provide minimum date

Minimum date:

☐ Provide maximum date

Maximum date:

Workspace2

- Clinical_Data_Mngt
- Code_Analysis
- HubRuns
- Install_Verification
- Migration
- ODS
- Pharmsug2023 (in HubExecution)
 - API
 - RunGenerateDates
 - email.zip
 - fail_email.sas
 - generate_dates.sas
 - success_email.sas
 - Reporting_Demo
 - sas2py
 - User_Defined_Functions
 - cardata.dat
 - fred.man

DESKTOP: Configure API Results Type

Altair Analytics Workbench

File Edit View Help

Log for Loc... write_read... regression... ODS_mod_tem... proc_reg_ex... *generate_d...

Hub Configuration

General Information

Label: RunGenerateDates

Program path: RunGenerateDates

Parameters

Results

type name filter text

Type	Name
Dataset	results

Add Remove

Result details

Name: results

Result type: Dataset

Dataset name: dates

Result format: JSON

XML

JSON

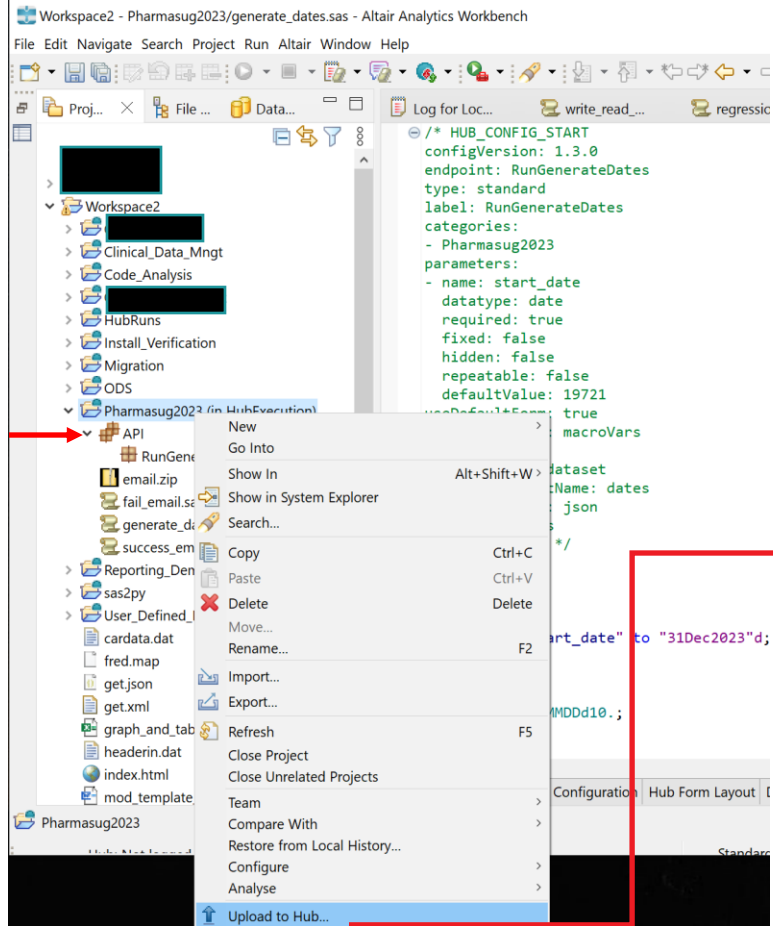
CSV

Categories

SAS Language Code Hub Configuration Hub Form Layout Description

DESKTOP: Upload API Package to Hub

© Altair Engineering Inc. Proprietary and Confidential. All rights reserved.



Workspace2 - Pharmasug2023/generate_dates.sas - Altair Analytics Workbench

File Edit Navigate Search Project Run Altair Window Help

Log for Loc... write_read... regression

Project Explorer:

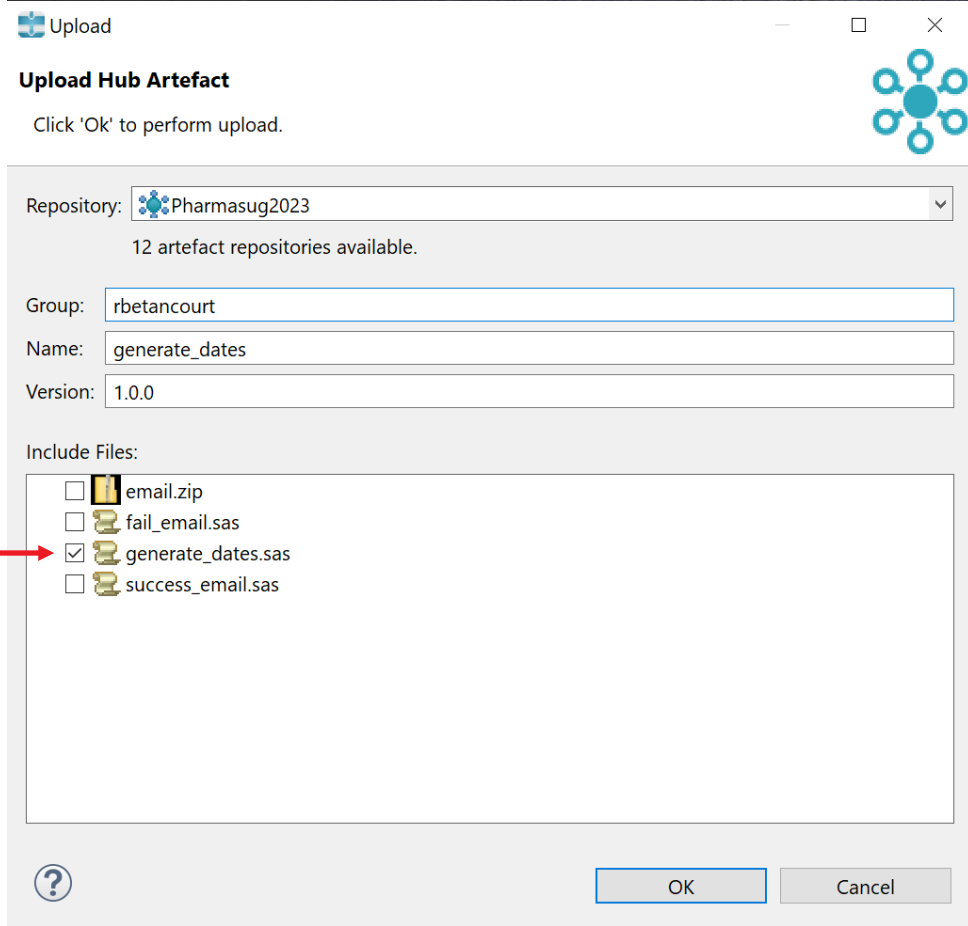
- Workspace2
 - Clinical_Data_Mngt
 - Code_Analysis
 - HubRuns
 - Install_Verification
 - Migration
 - ODS
 - Pharmasug2023 (in HubExecution)
 - API
 - RunGene
 - email.zip
 - fail_email.sas
 - generate_dates.sas
 - success_email.sas
 - Reporting_Den
 - sas2py
 - User_Defined
 - cardata.dat
 - fred.map
 - get.json
 - get.xml
 - graph_and_tab
 - headerin.dat
 - index.html
 - mod_template

Code Editor:

```
/* HUB_CONFIG_START
configVersion: 1.3.0
endpoint: RunGenerateDates
type: standard
label: RunGenerateDates
categories:
- Pharmasug2023
parameters:
- name: start_date
  datatype: date
  required: true
  fixed: false
  hidden: false
  repeatable: false
  defaultValue: 19721
  useDefaultValue: true
macroVars
dataset
Name: dates
json
*/
```

Context Menu (API folder):

- New
- Go Into
- Show In
- Show in System Explorer
- Search...
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Delete (Delete)
- Move...
- Rename... (F2)
- Import...
- Export...
- Refresh (F5)
- Close Project
- Close Unrelated Projects
- Team
- Compare With
- Restore from Local History...
- Configure
- Analyse
- Upload to Hub...



Upload

Upload Hub Artefact

Click 'OK' to perform upload.

Repository: Pharmasug2023

12 artefact repositories available.

Group: rbetancourt

Name: generate_dates

Version: 1.0.0

Include Files:

- ☐ email.zip
- ☐ fail_email.sas
- ☒ generate_dates.sas
- ☐ success_email.sas

Buttons: ? OK Cancel

HUB Portal: Select Artifact

© Altair Engineering Inc. Proprietary and Confidential. All rights reserved.

✓ Deployment Type

✓ Select artefact

3 Deployment path

4 Categories

Select artefact

Repository ⓘ

Pharmasug2023

Group ⓘ

rbetancourt

Artefact name ⓘ

generate_dates

Version ⓘ

1.0.0

Previous ←

Next →

✕ Cancel

HUB Portal: Define Deployment Path

© Altair Engineering Inc. Proprietary and Confidential. All rights reserved.

✓ Deployment Type ———— ✓ Select artefact ———— ✓ Deployment path ———— 4 Categories

Deployment path

Deployment path ⓘ generate_dates



Create API endpoint? ⓘ



Previous ←

Next →

✕ Cancel

HUB Portal: Make Category Selection

© Altair Engineering Inc. Proprietary and Confidential. All rights reserved.

✓ Deployment Type ———— ✓ Select artefact ———— ✓ Deployment path ———— ✓ Categories

Categories

Categories

Pharmasug2023 ✕

Type a category name and press enter, or select from dropdown list

Previous ←

Finish 

✕ Cancel

Altair SmartWorks Hub™

Artefacts

Artefact Repositories

Deployment

Asynchronous Jobs

Directory

Pipeline Runs

Pipelines

Deployment Services

Directory > RunGenerateDates

AboutExecute

Synchronous API endpoint

[https://\[REDACTED\]altair.com/run/generate_dates/RunGenerateDates](https://[REDACTED]altair.com/run/generate_dates/RunGenerateDates)

Asynchronous execution URL

[https://\[REDACTED\]altair.com/api/v2/ondemand/async/run/generate_dates/RunGenerateDates](https://[REDACTED]altair.com/api/v2/ondemand/async/run/generate_dates/RunGenerateDates)

Description

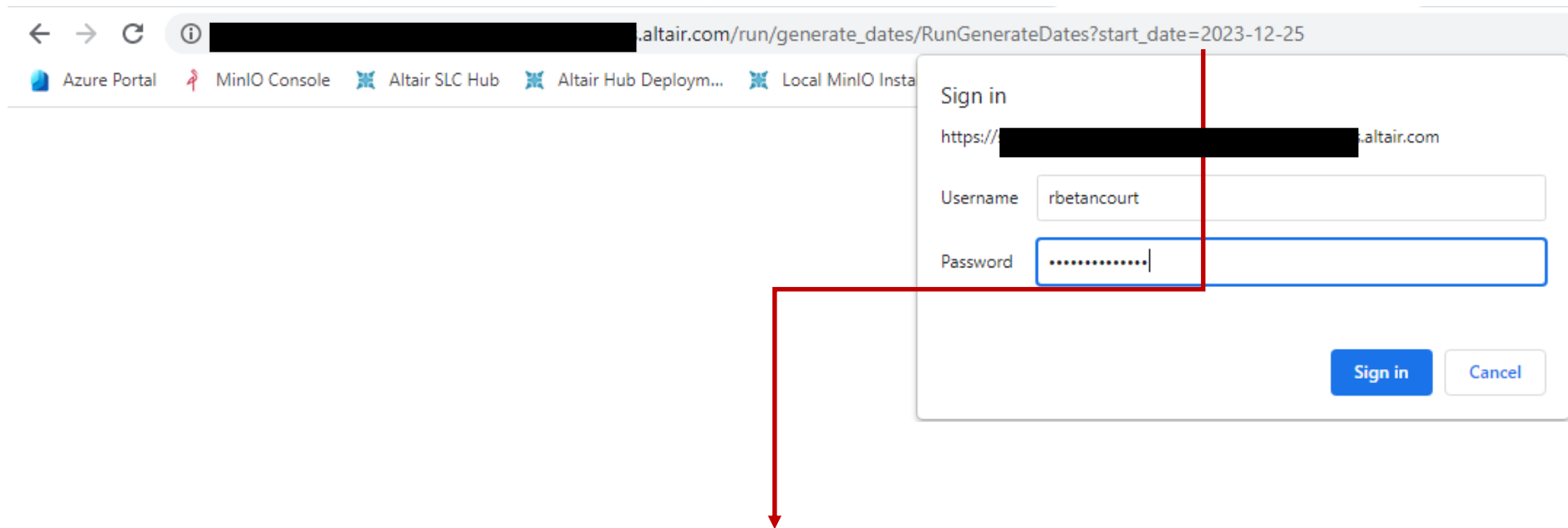
No description available

Parameters

Name	Type	Required
start_date	Date	✓

HUB Portal: Validate API Endpoint

© Altair Engineering Inc. Proprietary and Confidential. All rights reserved.



```
[{"date": "2023-12-25"}, {"date": "2023-12-26"}, {"date": "2023-12-27"}, {"date": "2023-12-28"}, {"date": "2023-12-29"}, {"date": "2023-12-30"}, {"date": "2023-12-31"}]
```

R Script Calls the SAS “DATES” REST URL

© Altair Engineering Inc. Proprietary and Confidential. All rights reserved.

```
# ChatGPT3 Prompt: write an R program to illustrate basic auth for httr library

auth_user <- "rbetancourt"

auth_pass <- "<HUB Password String>"

auth <- paste(auth_user, auth_pass, sep = ":")

# Encode the authentication details in base64 format

auth_enc <- enc2utf8(base64_enc(auth))

# Create a request with the authentication header

req <- GET("https://smartworkshub-
xxxxxxxxxx.xxxxxxxxxx.xxxxxxxxxxx.altair.com/run/generate_dates/RunGenerateDates?start_date=2023-
12-01",

add_headers("Authorization" = paste("Basic", auth_enc)))

# Extract the JSON content from the response

json_content <- content(req, "text")

# Convert the JSON content to a DataFrame

df <- fromJSON(json_content)

print(df)
```



R Results from calling SAS Language API

© Altair Engineering Inc. Proprietary and Confidential. All rights reserved.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

No encoding supplied: defaulting to UTF-8.

```
date
1 2023-12-01
2 2023-12-02
3 2023-12-03
4 2023-12-04
5 2023-12-05
6 2023-12-06
7 2023-12-07
8 2023-12-08
9 2023-12-09
10 2023-12-10
11 2023-12-11
12 2023-12-12
13 2023-12-13
14 2023-12-14
15 2023-12-15
16 2023-12-16
```

Install P... ✕

R Interactive