# Using The Postman® & SoapUI® Application with Sage X3

Raheel Khan

# Contents

## SoapUI® Application

- Sage X3 Prerequisites for SOAP Web Services
- Introduction to the SoapUI® Application
- Installation of SoapUI®
- The Sage X3 SOAP WSDL
- How to send a request to X3
    - Loading the X3 WSDL
    - Importing & Saving Projects
- Load Testing with SoapUI®
    - Creating a test case
    - Defining data and assertions
    - Configuration

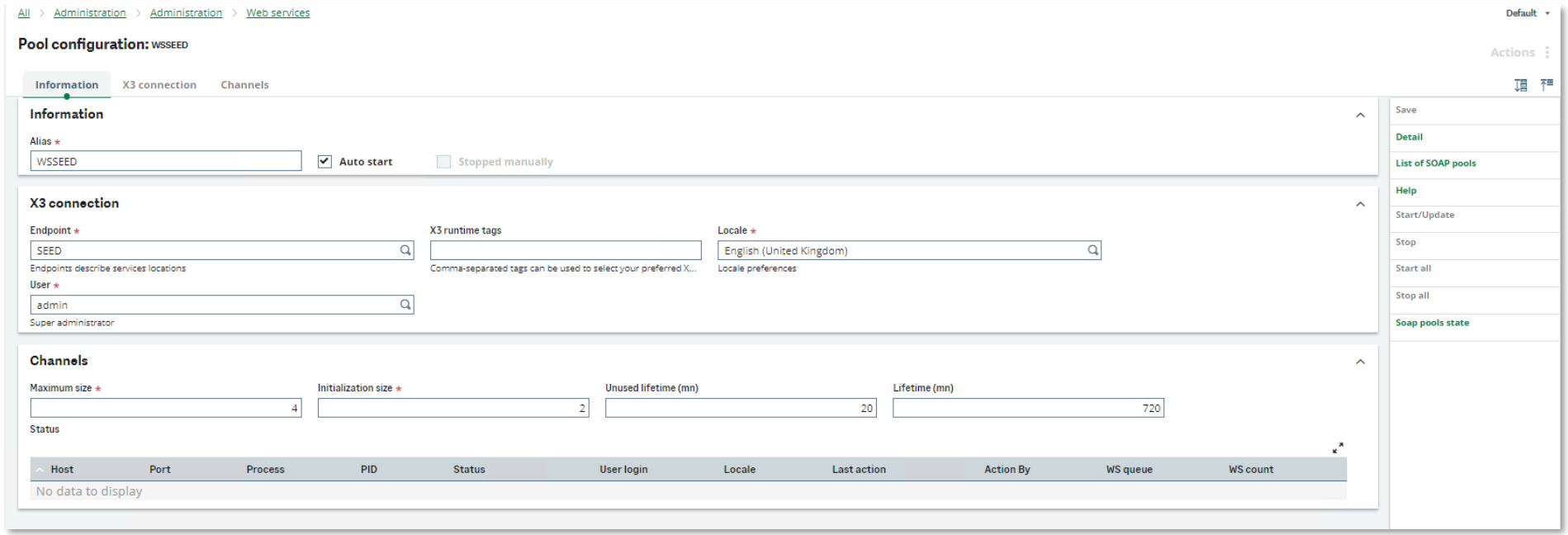
## Postman Application

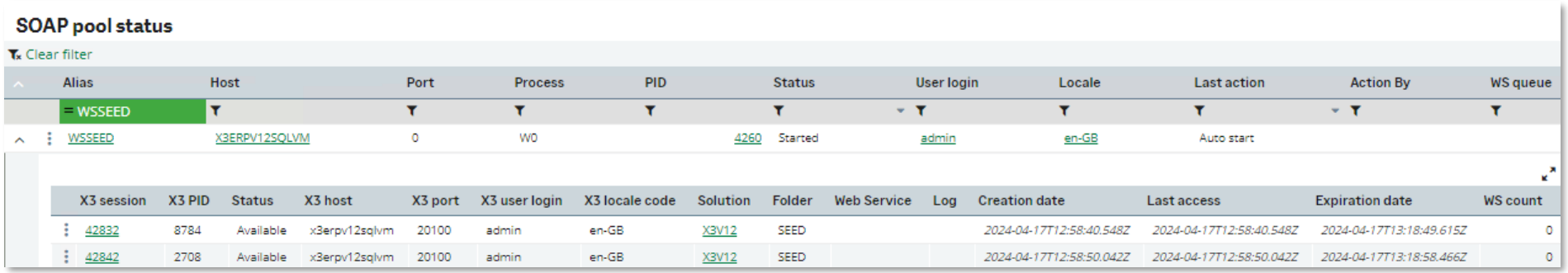
- Sage X3 and Graph QL API
- Introduction to the Postman Application
- Installation of Postman
- How to send a Request to X3 using Postman
    - Authentication
    - SOAP Request
    - Graph QL Request
- Tips
    - Request Tips
    - Troubleshooting when no WS response
    - Graph QL Network inspector

# Sage X3 Prerequisites for SOAP

# Sage X3 Prerequisites for SOAP

SOAP Web pool needs to be configured and channels available so that the web service is available externally.

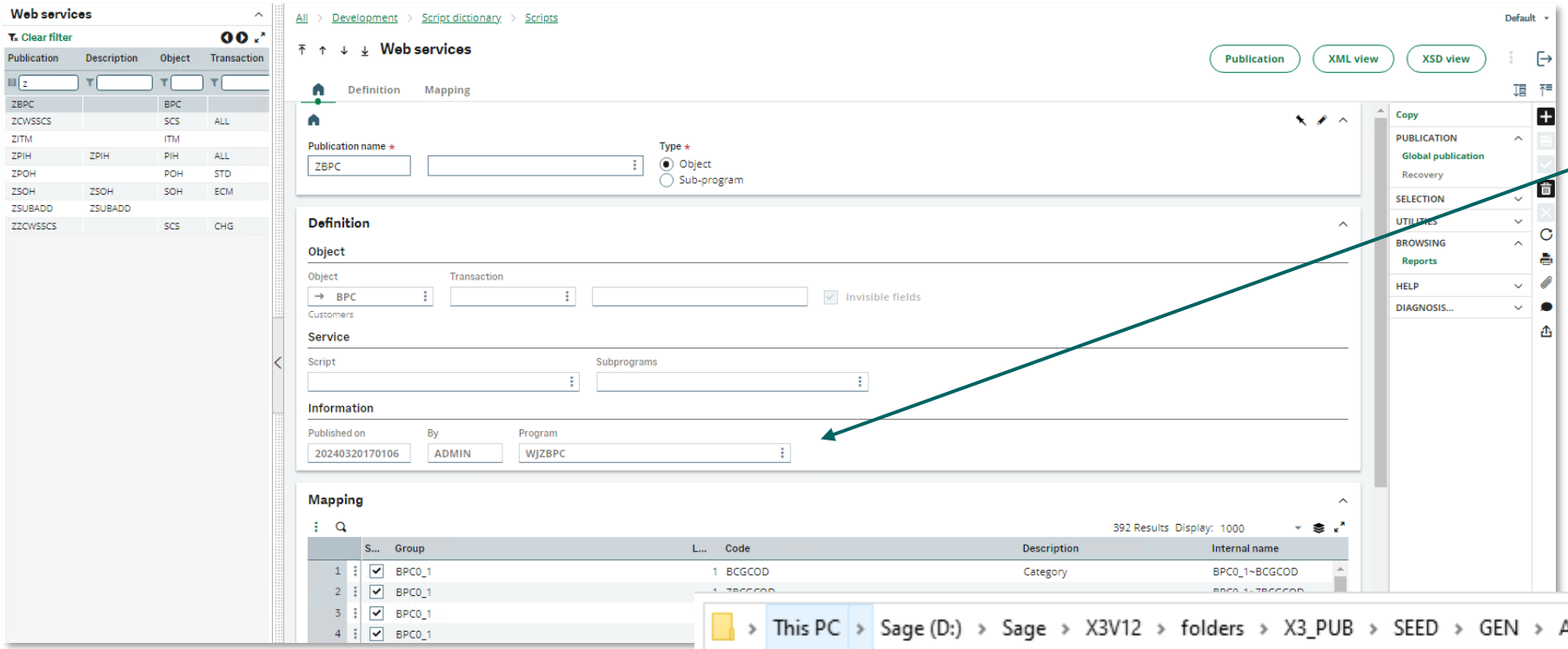


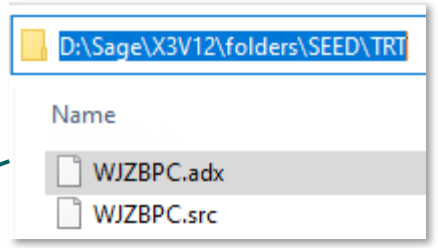




- Alias that identifies the pool
- Auto start to start the pool automatically when Sage X3 starts
- Default startup language
- Default startup Sage X3 user code
- Max size (maximum number of connections)
- Initial size (number of connections at startup).

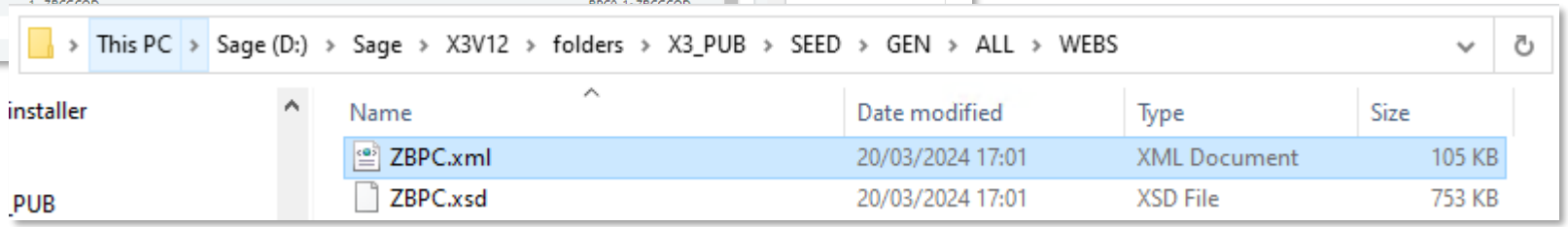
# Sage X3 Prerequisites for SOAP

**Web service needs to be published** for the Web services a simple function exists under Scripts in the Development > Script dictionary > Web Services GESAWE to transform a subprogram or object into a Web service with a public name.



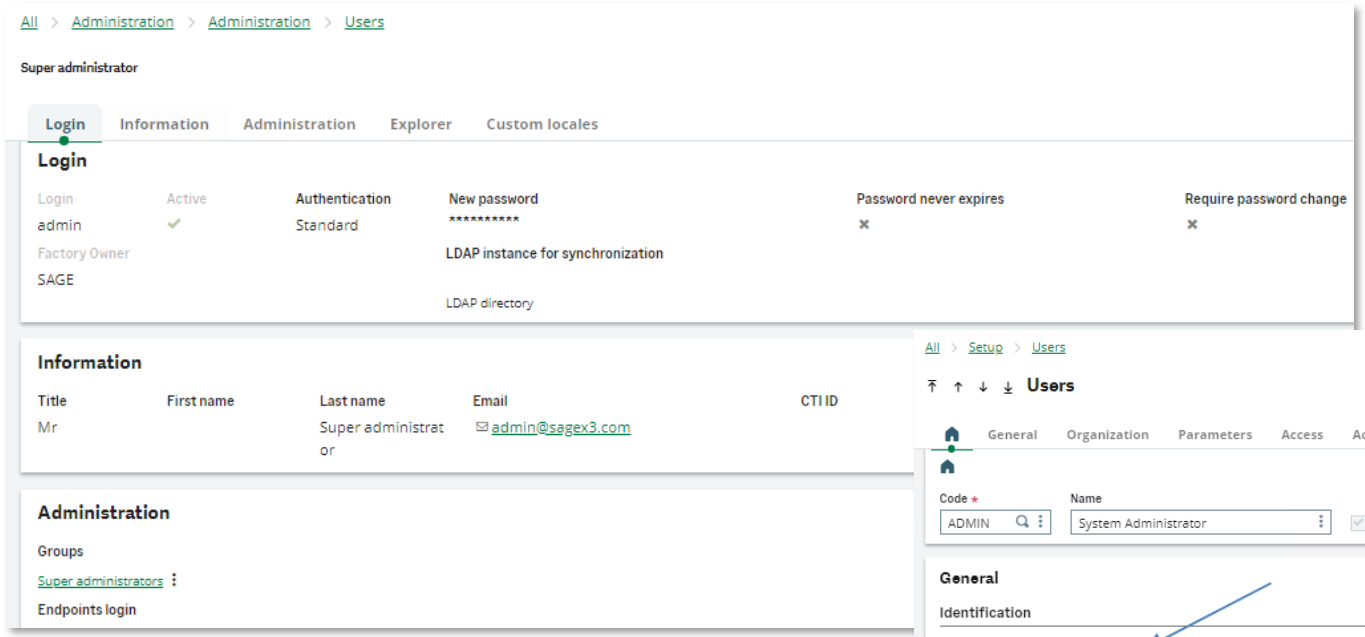




A XML description, stored in the following directory on the applications server: X3_PUB/FolderName/GEN/ALL/WEBS is generated

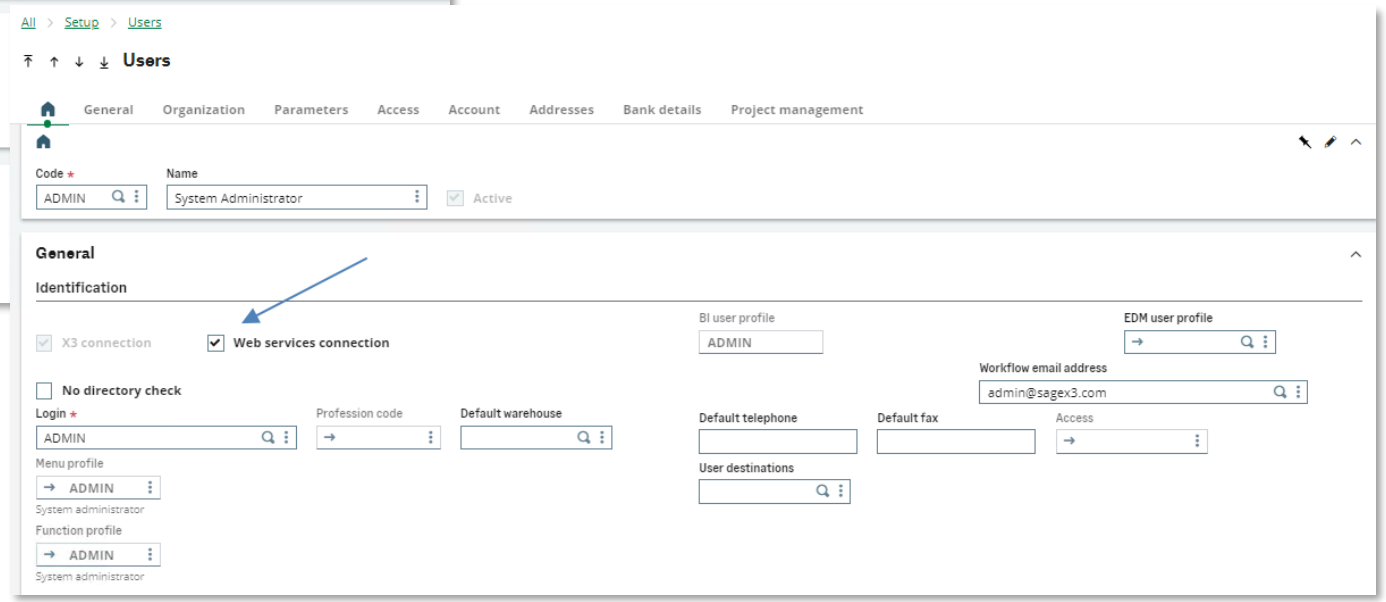


# Sage X3 Prerequisites for SOAP

A user needs to be configured for authentication




Syracuse user setup




Sage X3 Classic  user setup with web services connection allowed

# SoapUI® Application

# What is the SoapUI® Application

SoapUI® by SmartBear is a widely used open-source tool for testing SOAP (Simple Object Access Protocol) and REST (Representational State Transfer) web services. It was first launched in 2006.

**Key Features:**

**Versions:** SoapUI is available in both open-source and commercial versions. The open-source version offers robust features for most testing needs, while the commercial version provides additional enterprise-level features and support.

**Supported Technologies:** SoapUI supports various technologies, including SOAP, REST, HTTP, JMS, AMF, JDBC, and more, making it versatile for testing a wide range of APIs and web services.

•**Functional Testing:** SoapUI allows users to create and execute functional tests for SOAP and REST APIs.

•**Automated Testing:** It supports automated testing with features like assertions, data-driven testing, and scriptable testing scenarios.

•**Load Testing:** It provides load testing capabilities to assess the performance and scalability of web services under various conditions.

•**Mocking:** SoapUI enables the creation of mock services, allowing developers to simulate APIs and test client applications without accessing the service.

•**Security Testing:** SoapUI supports security testing by configuring security protocols and conducting vulnerability assessments.

# SoapUI® With Sage X3

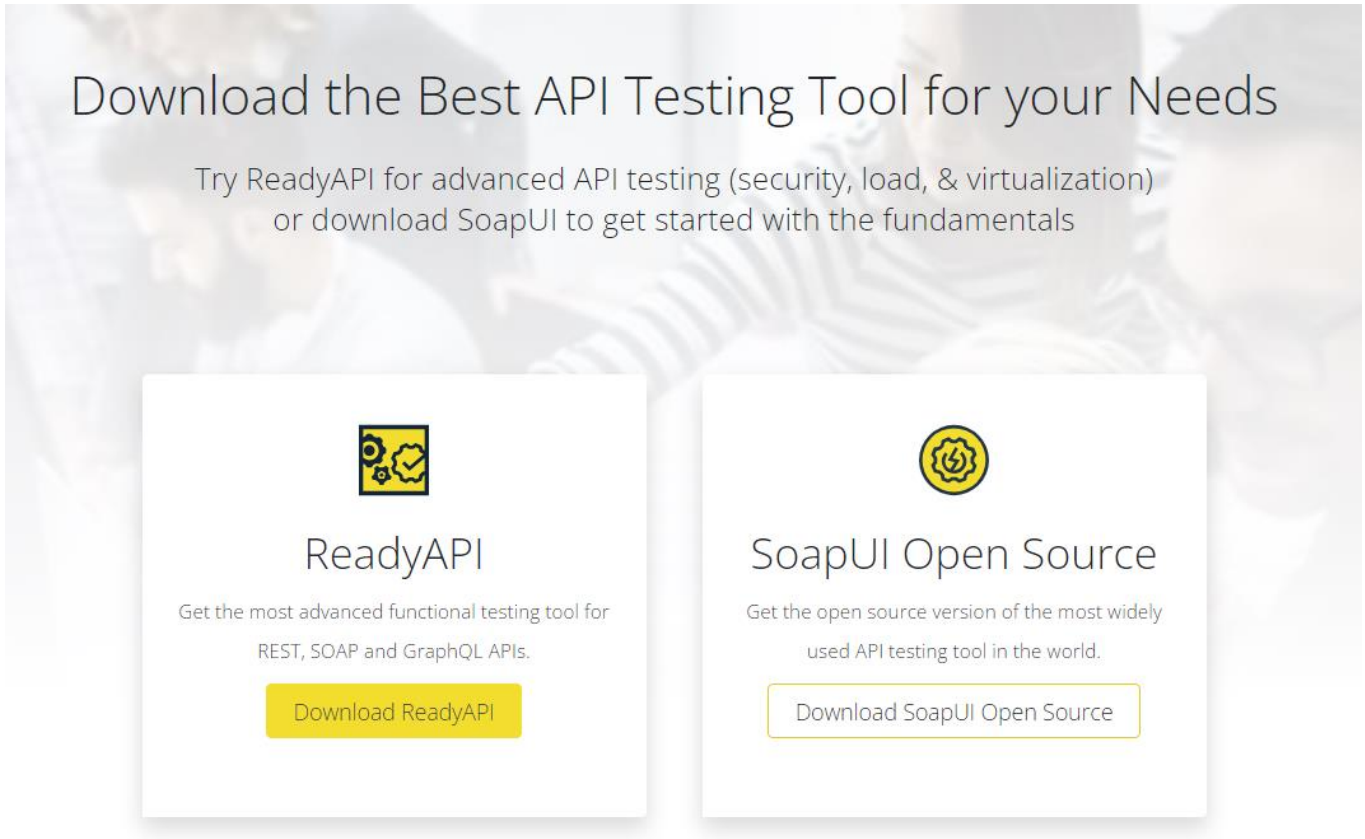
SoapUI can be used with Sage X3

1. Using a Projects in SoapUI to save your requests
2. Testing SOAP web services outside side of Sage X3
3. Testing payloads that may not be suitable for the internal tester
4. Automated testing
5. Load testing

# Installing SoapUI® Application

# SoapUI® Application Download

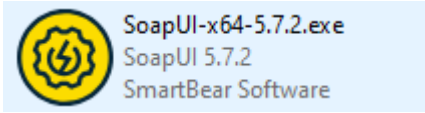
SoapUI application can be downloaded from https://www.SoapUI.org/downloads/SoapUI/



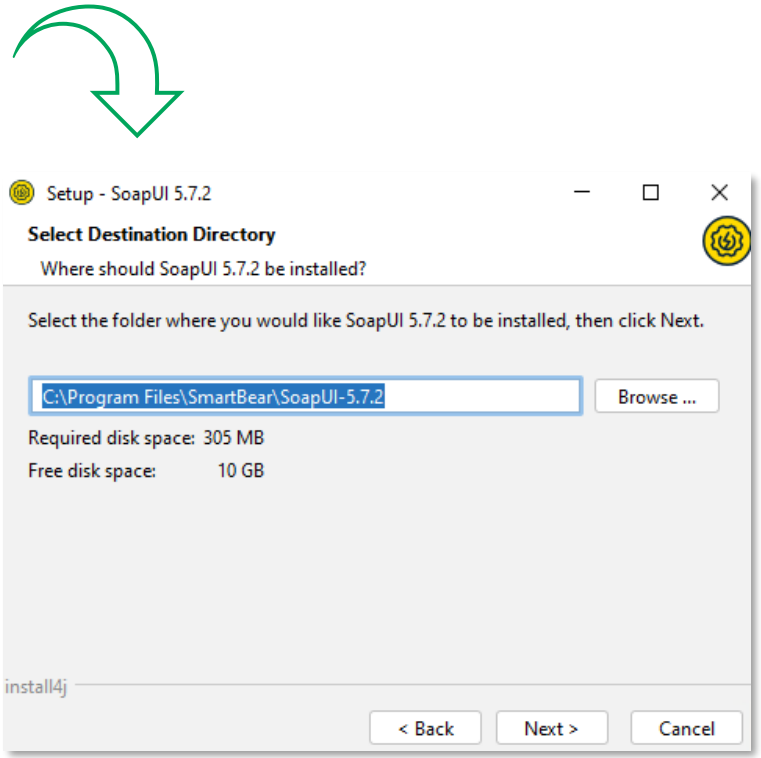

There are two versions available for Windows, Linux, MacOS

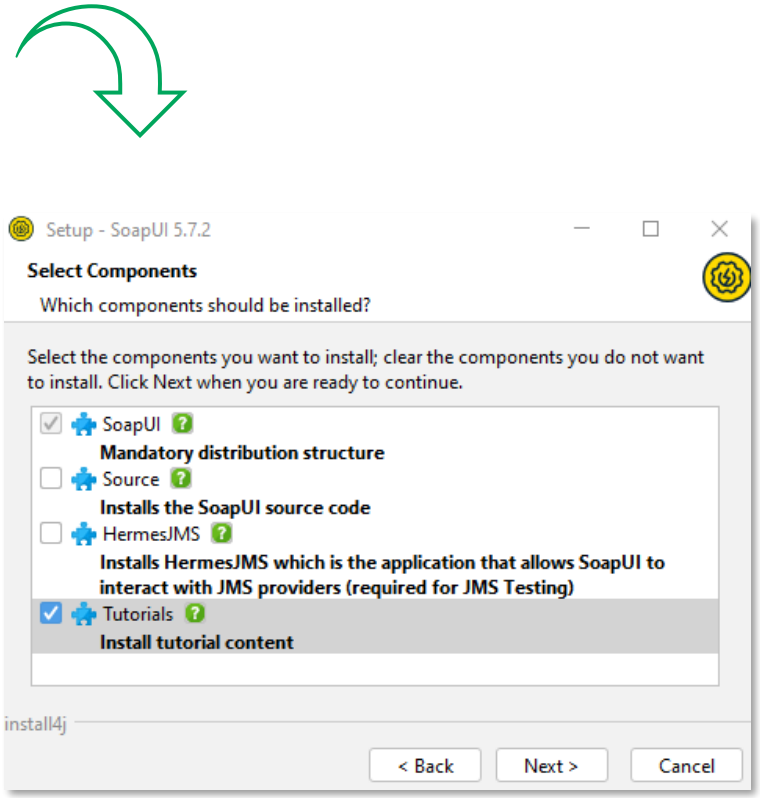
Ready API is the commercial version that has extra functionality

- Improved interface
- Team collaboration features (Jira, GitHub, Slack etc)
- More advanced testing features
- Pre-built reporting features



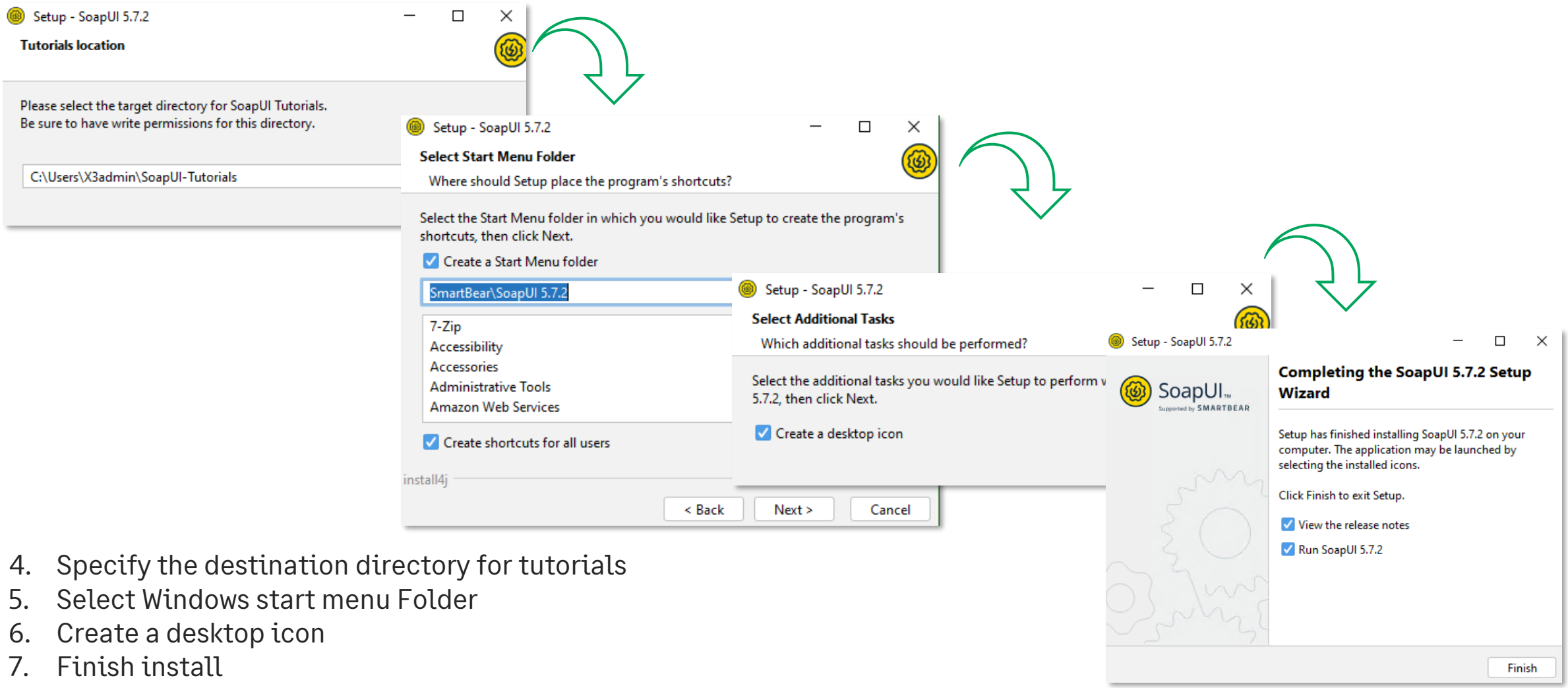
# Installing SoapUI® Application


1. Execute the.exe file for Windows
2. Choose the installation directory
3. Choose components

# Installing SoapUI® Application



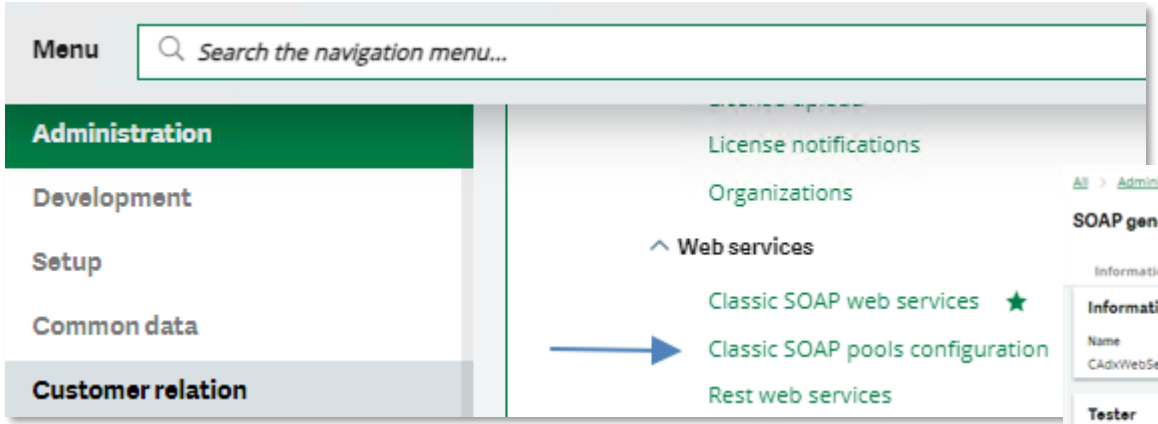

4. Specify the destination directory for tutorials
5. Select Windows start menu Folder
6. Create a desktop icon
7. Finish install

# Sage X3 SOAP WSDL

# The Sage X3 Soap WSDL

A SOAP (Simple Object Access Protocol) WSDL (Web Services Description Language) is a document written in XML that describes the functionalities offered by a SOAP-based web service.

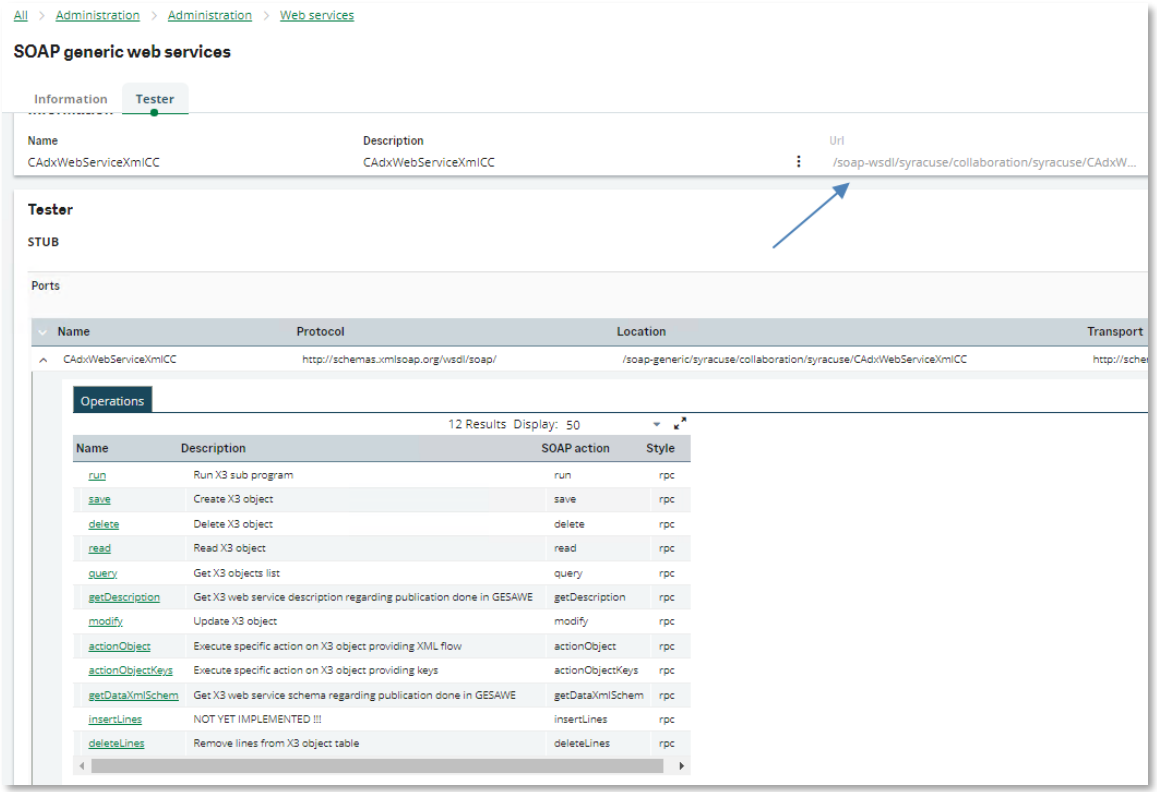
WSDL defines the methods available, the parameters required for each method, the data types used, and the communication protocols (typically HTTP and HTTPS) supported by the web service. It acts as a contract between the service provider, in our case Sage X3, and the service consumer (External application), enabling communication between different systems and platforms.



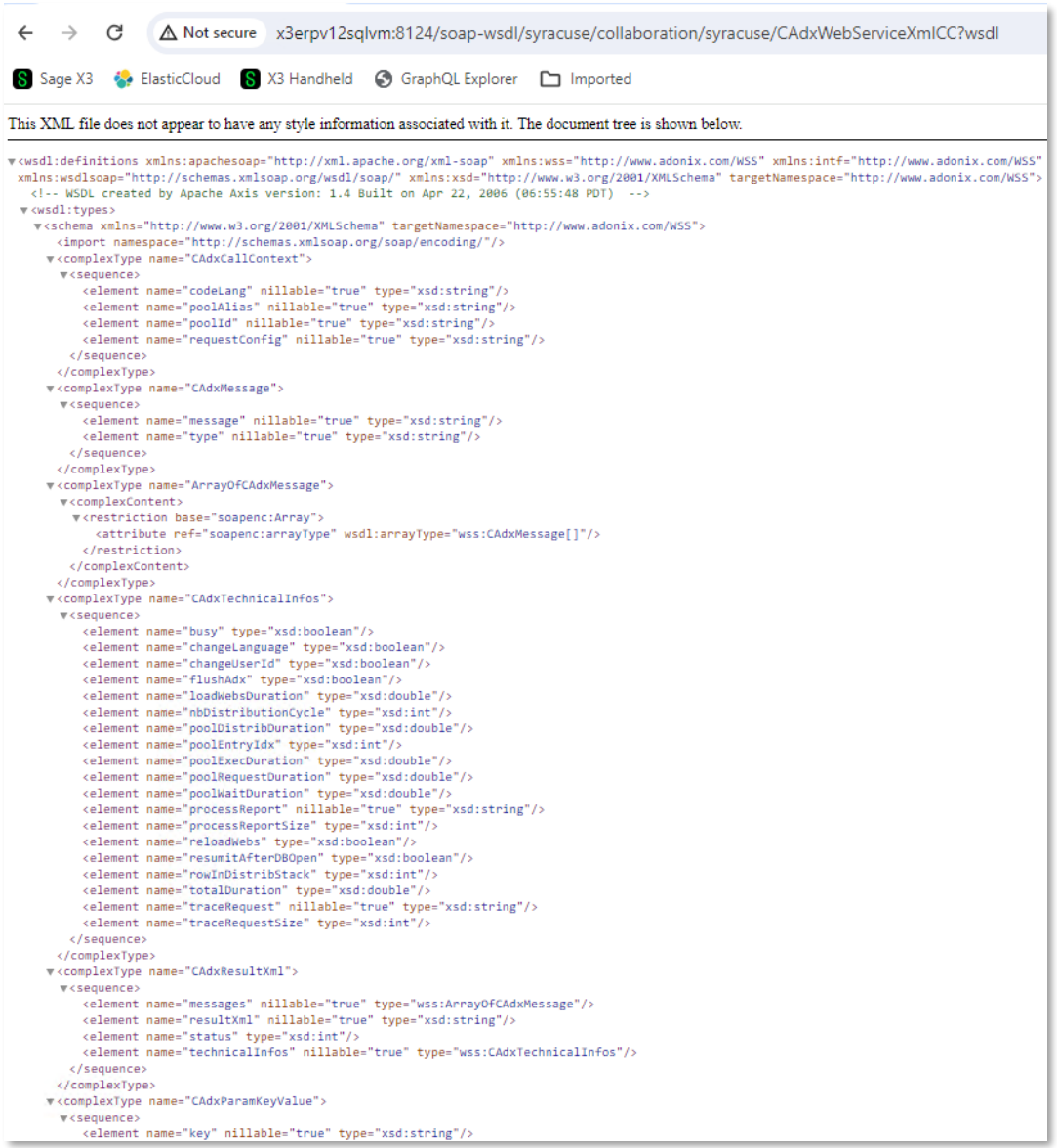

Access the SOAP WSDL from the Classic SOAP pools configuration function

# The Sage X3 Soap WSDL

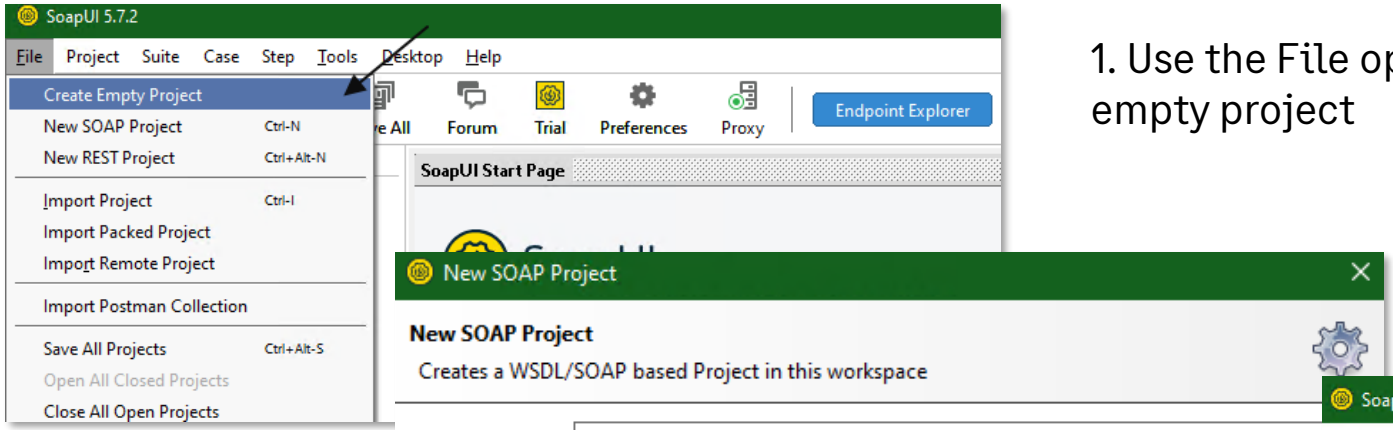
When accessing the URL you can see the XML definition for the WSDL, Sage also has an internal SOAP testing utility which can be accessed from the same function



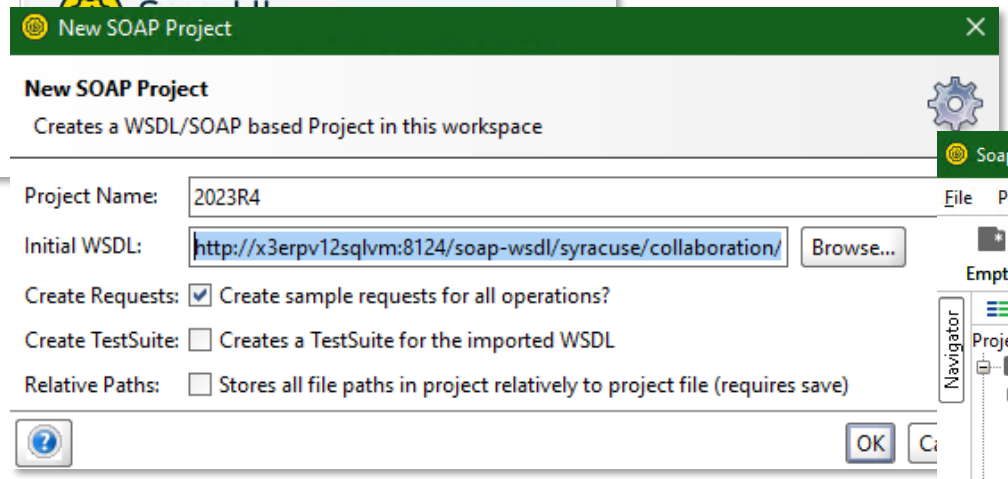

**URL**
http://x3erpv12sqlvm:8124/soap-wsdl/syracuse/collaboration/syracuse/CAdxWebServiceXmlCC?wsdl



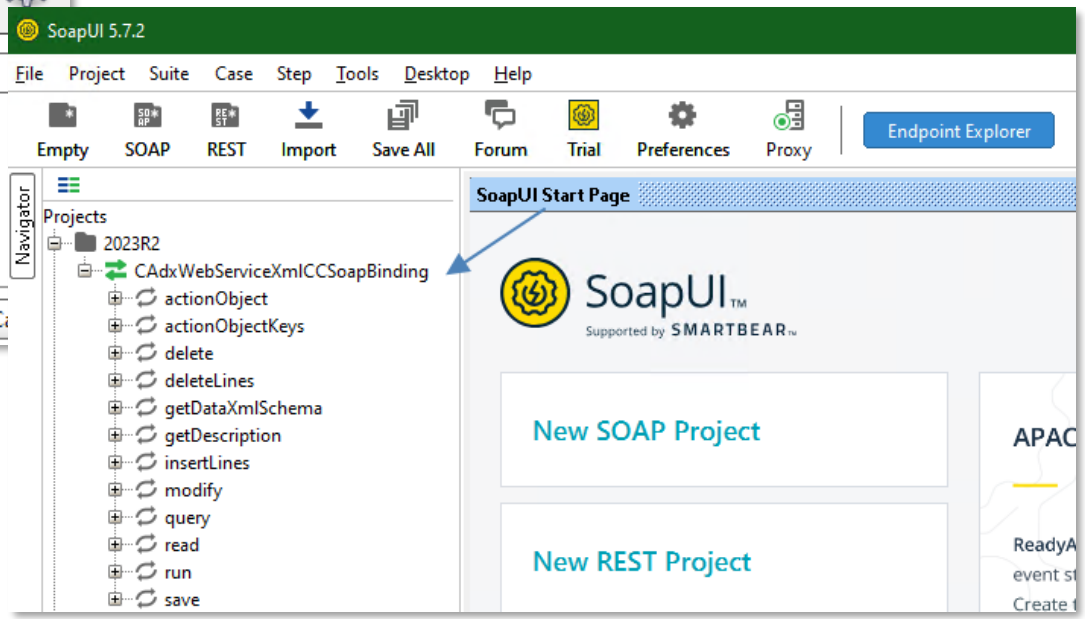
# Loading The Sage X3 SOAP WSDL




1. Use the File option to create a new empty project



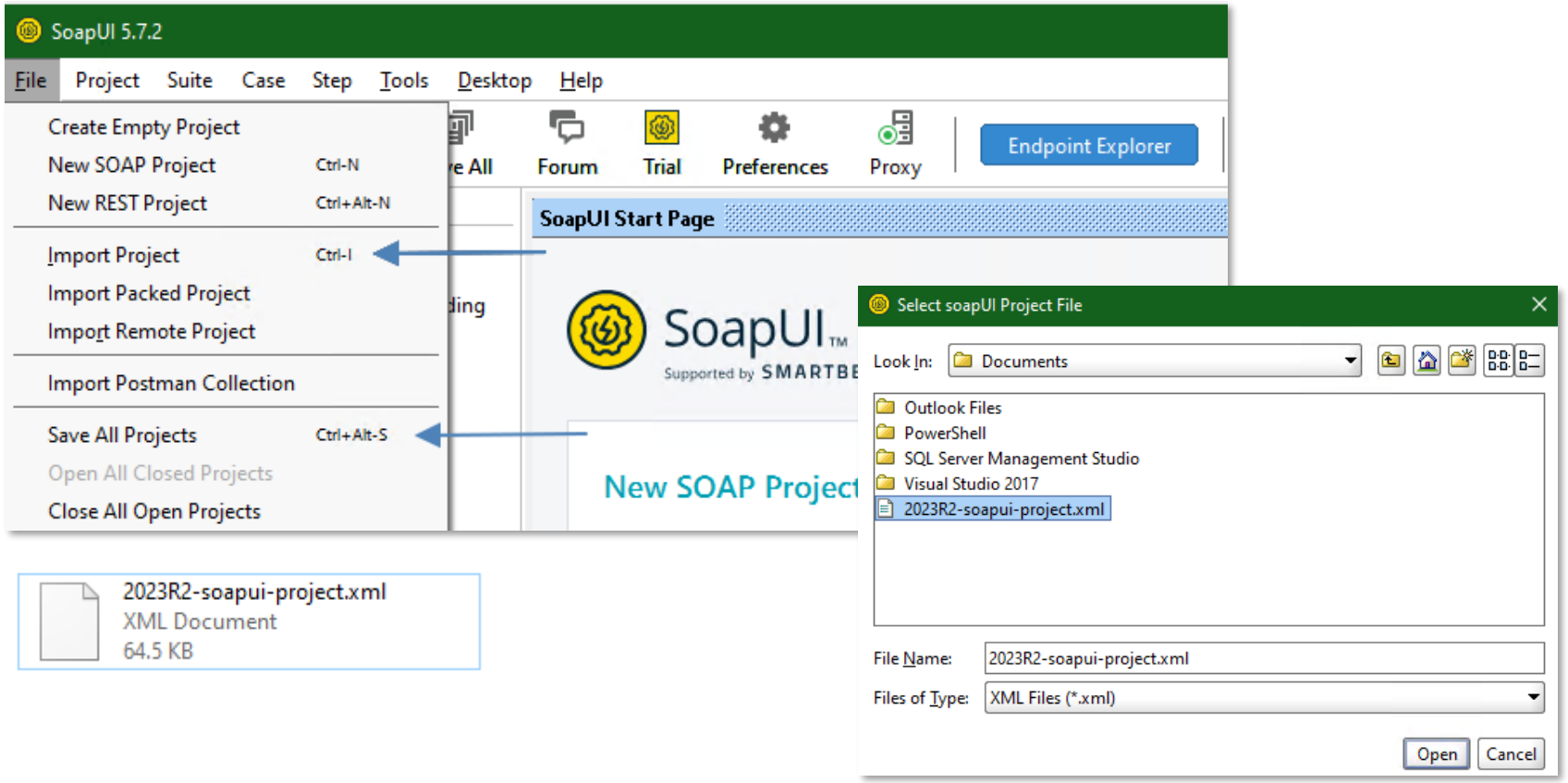

2. Enter a project name & Populate the WSDL URL




3. SoapUI® creates the new project and loads the WSDL methods

# Loading & Saving Your Project

SoapUI® Projects are saved in XML format




- When you open SoapUI®, existing projects are auto-loaded
- Use ‘Import project’ if you want to load an existing project from an XML file
- Use ‘Save All’ to save any changes to your project

Note that the project file will include any passwords/authentication details that are used in the project

# Sending Requests Using SoapUI®

# Sending SOAP Request To Sage X3

To send a SOAP request to Sage X3 using SoapUI®

- **Create your SOAP request** in the correct request category for example ‘Read’ or ‘Query’

Web service class/method

Name of the web service pool

Web service public name

Language code (optional)

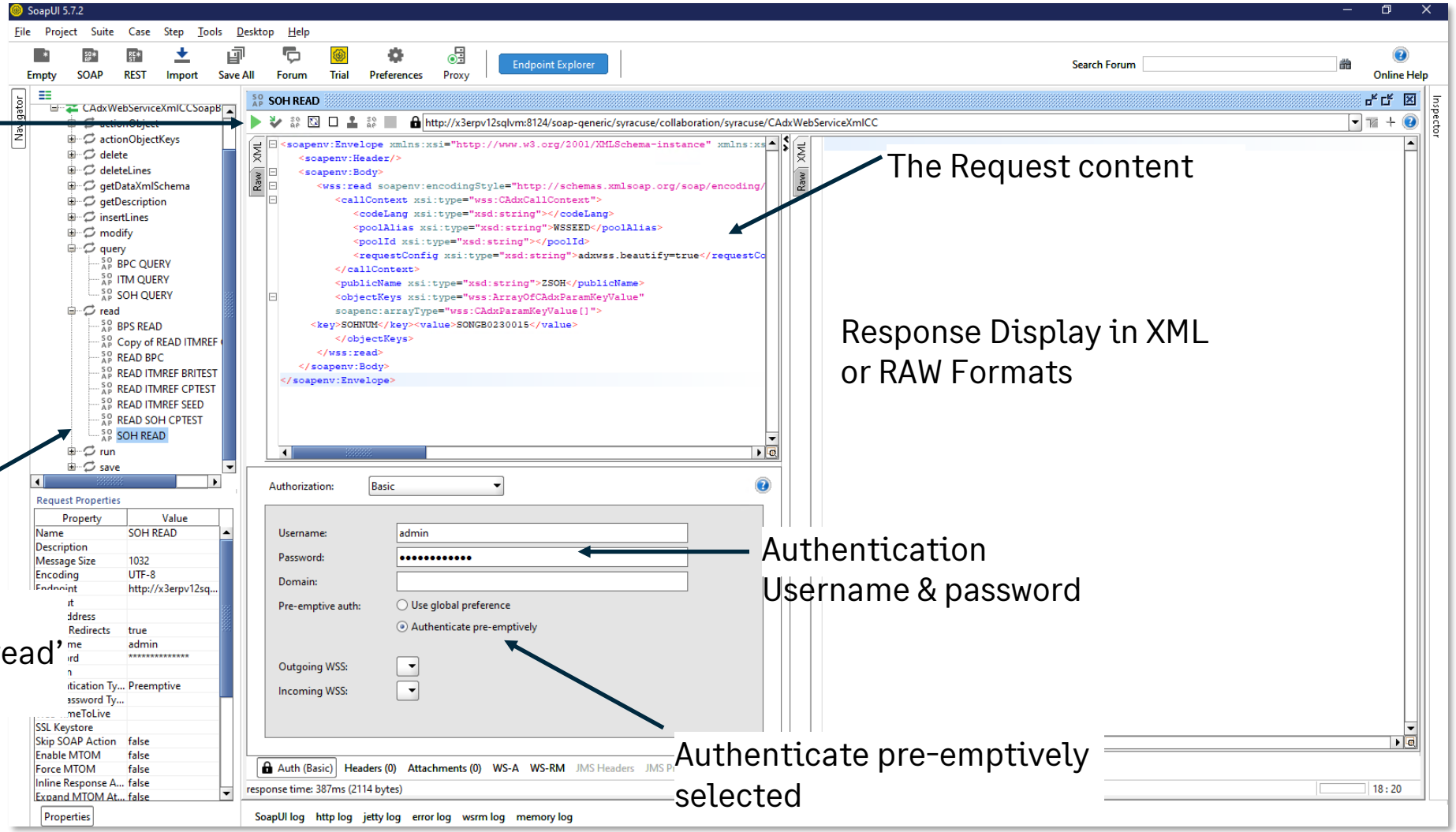
Object key (for Read, Modify and Delete methods). It has two public properties: key (the Sage X3 field name or rank in the main index) and Value (the key value

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wss="http://www.adonix.com/WSS"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <soapenv:Header/>
  <soapenv:Body>
    <wss:read soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <callContext xsi:type="wss:CAdxCallContext">
        <codeLang xsi:type="xsd:string">BRI</codeLang>
        <poolAlias xsi:type="xsd:string">WSSEED</poolAlias>
        <poolId xsi:type="xsd:string"></poolId>
        <requestConfig xsi:type="xsd:string">adxwss.beautify=true</requestConfig>
      </callContext>
      <publicName xsi:type="xsd:string">ZSOH</publicName>
      <objectKeys xsi:type="wss:ArrayOfCAdxParamKeyValue" soapenc:arrayType="wss:CAdxParamKeyValue[]">
      <key>SOHNUM</key><value>SONGB0230015</value>
      </objectKeys>
    </wss:read>
  </soapenv:Body>
</soapenv:Envelope>
```

- **Set your authentication settings** to authenticate pre-emptively (pre-emptive authentication refers to the process of sending authentication credentials with every request without waiting for the server to challenge the client. Unlike the standard authentication process where the client waits for the server to respond with a challenge before sending the credentials)

# Sending SOAP Request To Sage X3

Submit request




The Request content

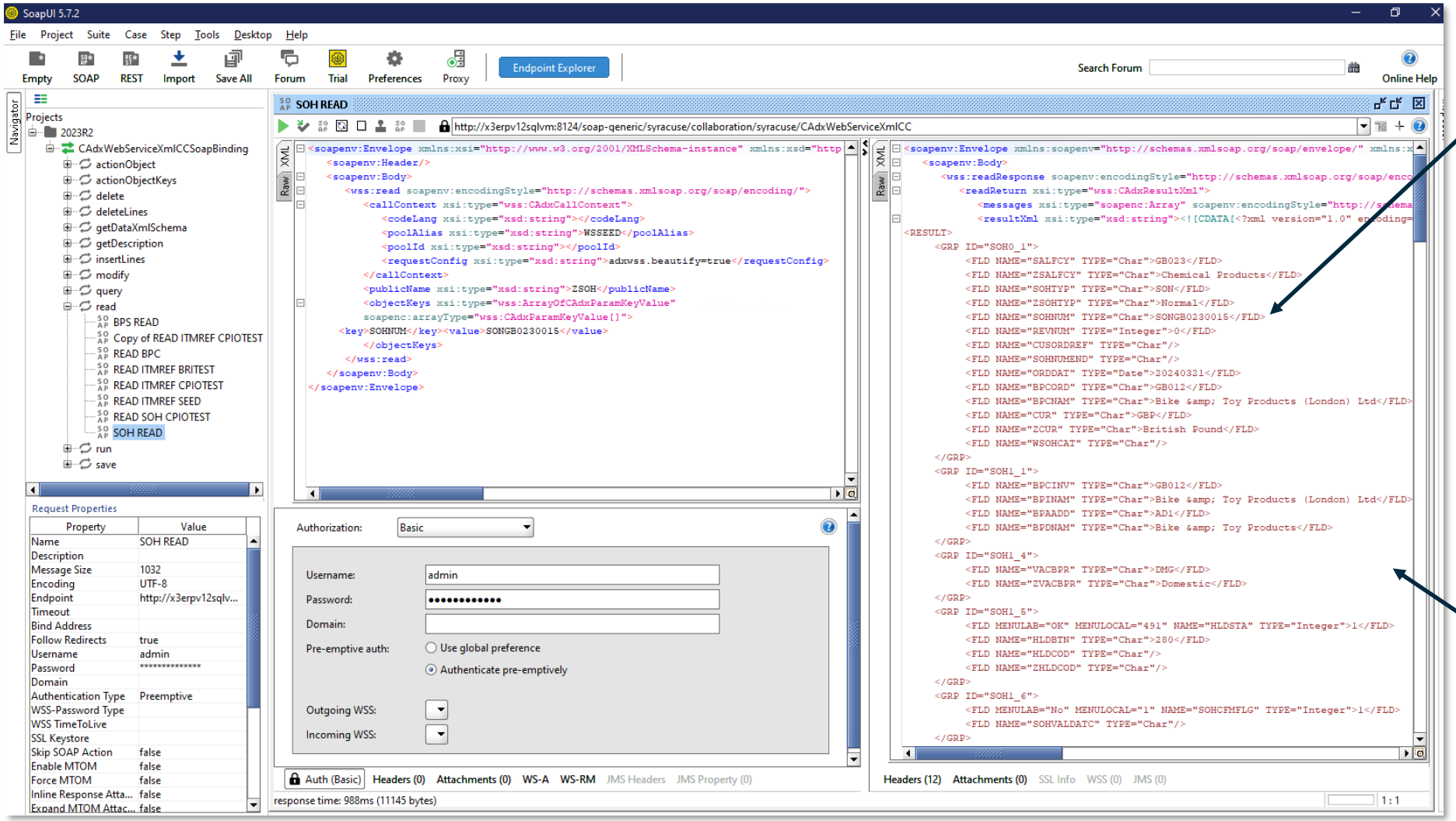
Response Display in XML
or RAW Formats

Request SOH Read
created under the ‘read’
method

Authentication
Username & password

Authenticate pre-emptively
selected

# Receiving SOAP Response From Sage X3



# Load Testing With SoapUI®

# Load Testing With SoapUI®

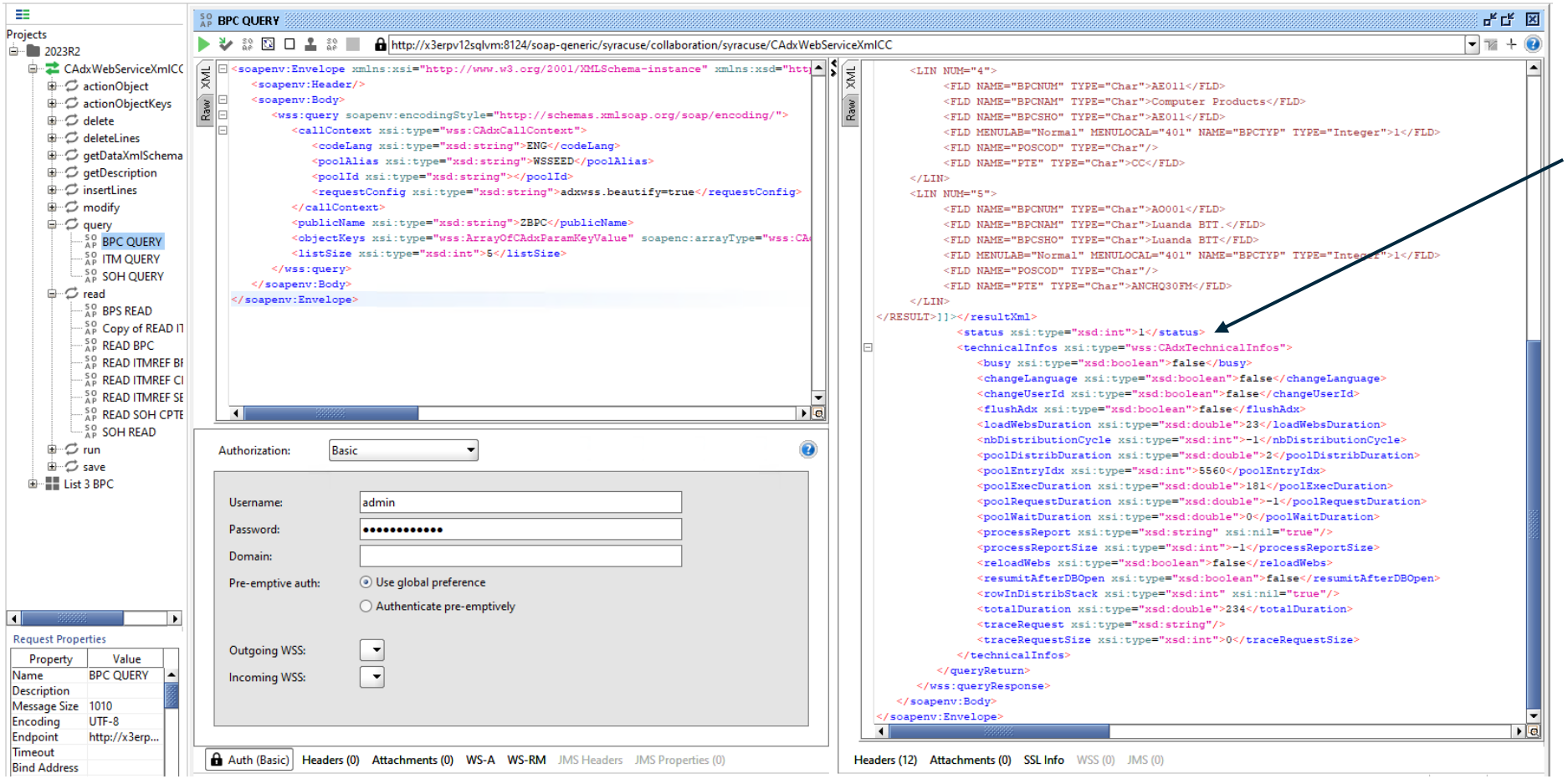
Load testing in SoapUI® refers to the process of simulating multiple users accessing a web service to assess its performance under various load conditions. This can help to identify bottlenecks, determine system behaviour under peak load, and ensure that the service can handle the expected number of users without degradation in performance.

**The main steps to create a simple load test in SoapUI®**

1. Creating the test case
2. Define Test Data and Assertions
3. Configure Load Test Settings
    1. Threads (virtual users)
    2. Set the total number of requests or time to run requests
    3. Set strategy
    4. Execute the test and review results

# Creating Test Case

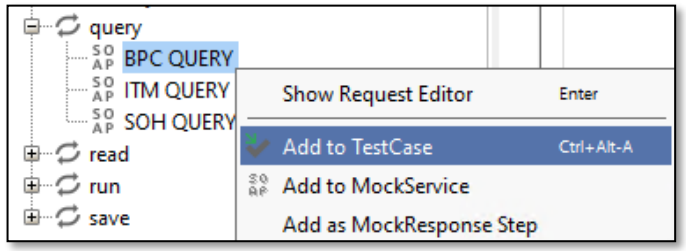
A test case typically contains a sequence of requests or a single request to be executed during the load test. For example, we have a SOAP request which uses the query method to list 5 customer records.



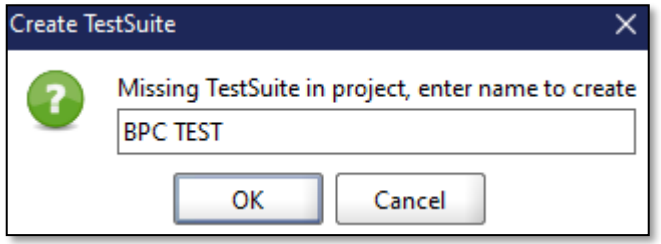

Testing the query response returns the correct results status 1 is returned = successful request & response

# Define Test Data

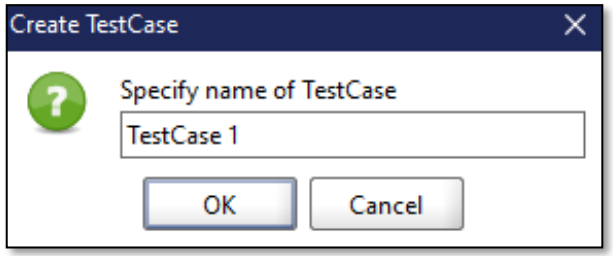
Our test data will be a single SOAP request and response. Next, we need to set up the assertions within the test case to validate the correctness of the responses during the lad test.



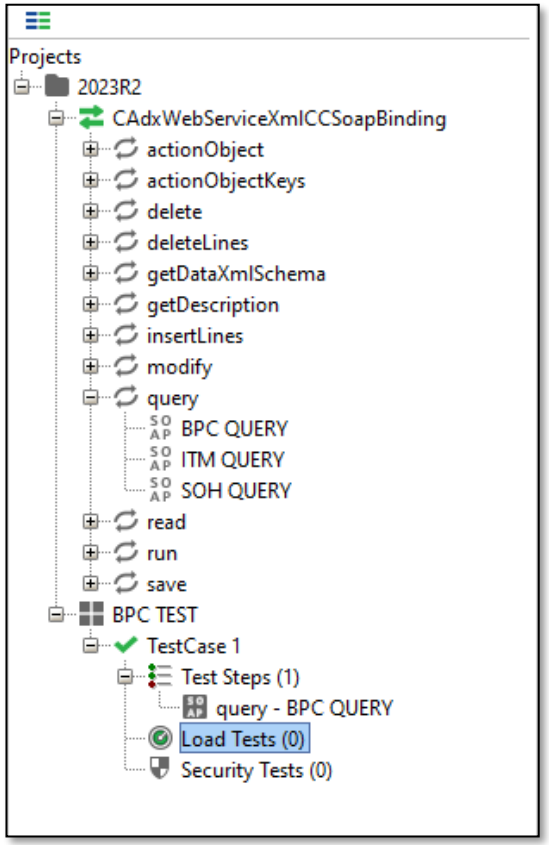

Right-click on the query and select add to test case




Create a test suite in the current project.



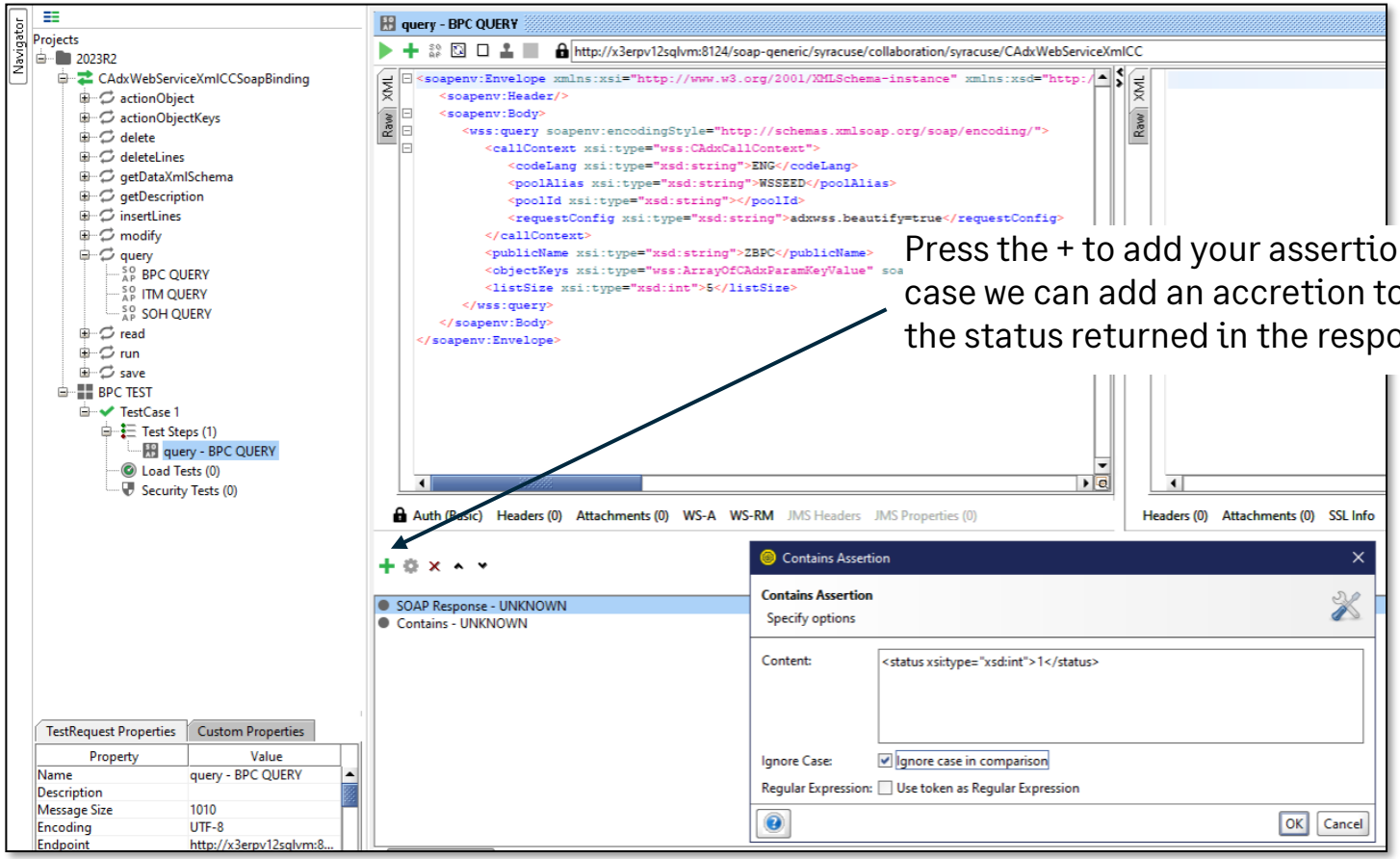

Give your test case a name




The test suite is created with the selected request.

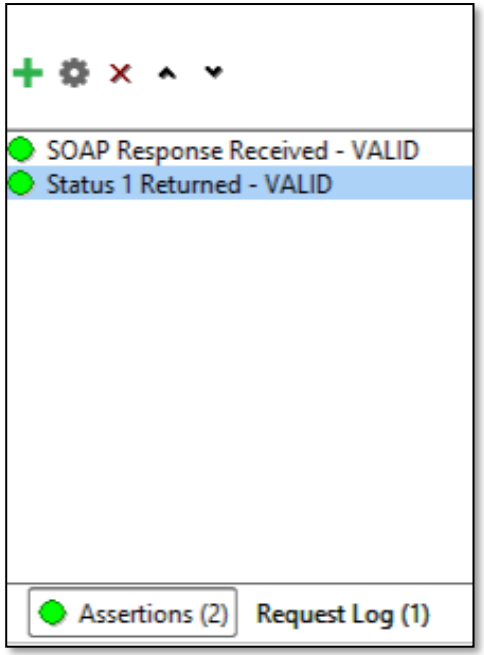
# Define The Assertions

Assertions in SoapUI are validation checkpoints that allow you to verify the correctness of responses received from web services during testing. They are used to define criteria that must be met for a test step or a test case to pass. Assertions help ensure the web service behaves as expected and meets the specified requirements.

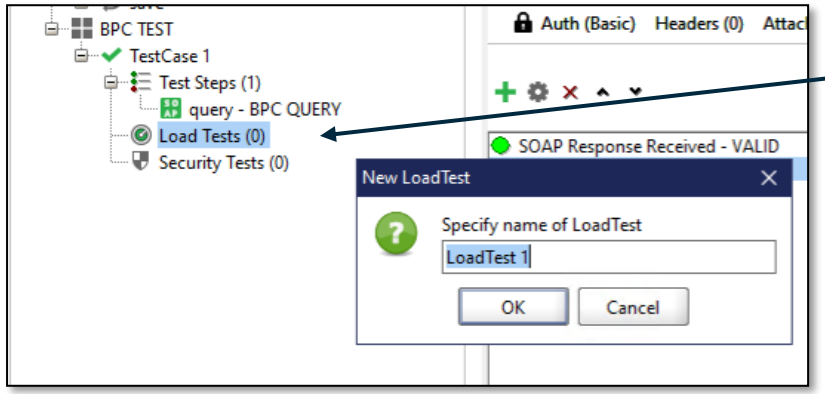



Press the + to add your assertions in this case we can add an accretion to check the status returned in the response is '1'

Press play to launch the query and confirm the assertion status is working

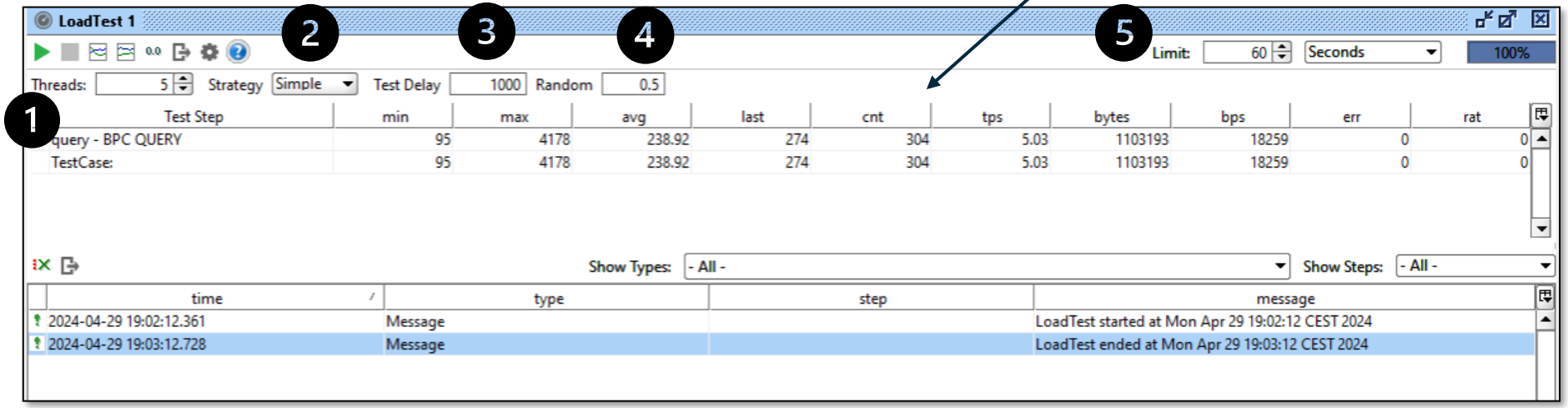


# Create/Configure Load Test Settings




The last step is creating the load test and then configuring the test parameters. Right-click on load tests and select the option to create a new test giving it a name.

Cnt is the number of requests sent and if any of our defined assertions fail are logged in the err column




1. Number Threads: Specify how many simultaneous users should be simulated during the test.
2. The different test strategies allow you to configure different test options simple making available test delay and randomisation
3. Test delay sets the delay between each test run in milliseconds
4. Random amount of randomisation in the delay 0 = no randomisation
5. Limit sets the total number of tests or time in seconds for the test to run

# Postman ® Application

# What Is The Postman Application

Postman is an API (Application Programming Interface) development and testing tool that simplifies the process of building, testing, and documenting APIs. It provides an intuitive user interface for creating and sending HTTP requests to APIs, allowing developers to interact with APIs without writing code manually.

**Key features of Postman include:**

**Request Building:** Postman allows users to easily create HTTP requests such as GET, POST, PUT, DELETE, etc., and customize request headers, parameters, and payloads.

**Collection Management:** Users can organise requests into collections, making managing and sharing related API endpoints and test scenarios easier.

**Automated Testing:** Postman supports the creation of automated test scripts using JavaScript. Developers can write tests to verify API responses, status codes, and data integrity.

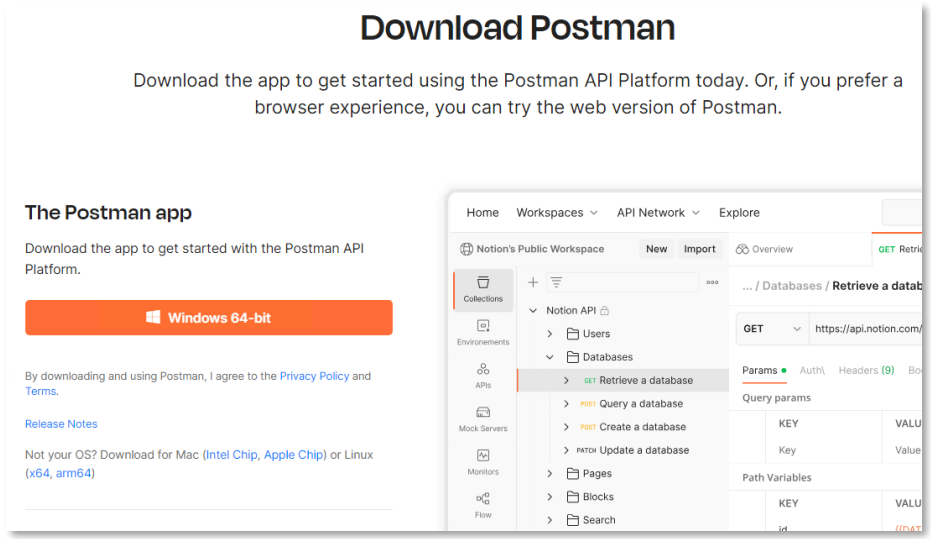
**API Documentation**: Postman enables the generation of interactive API documentation from API requests and responses, making it easier for developers to understand and use APIs.

**Collaboration:** Postman provides features that allow team members to share collections, collaborate on API development, and work together on API testing and debugging.

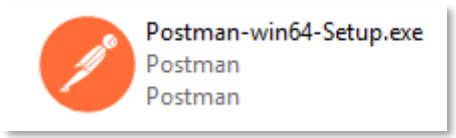
Overall, Postman streamlines the API development lifecycle by providing a comprehensive set of tools for designing, testing, and debugging APIs, making it a valuable tool for developers and teams working with APIs.

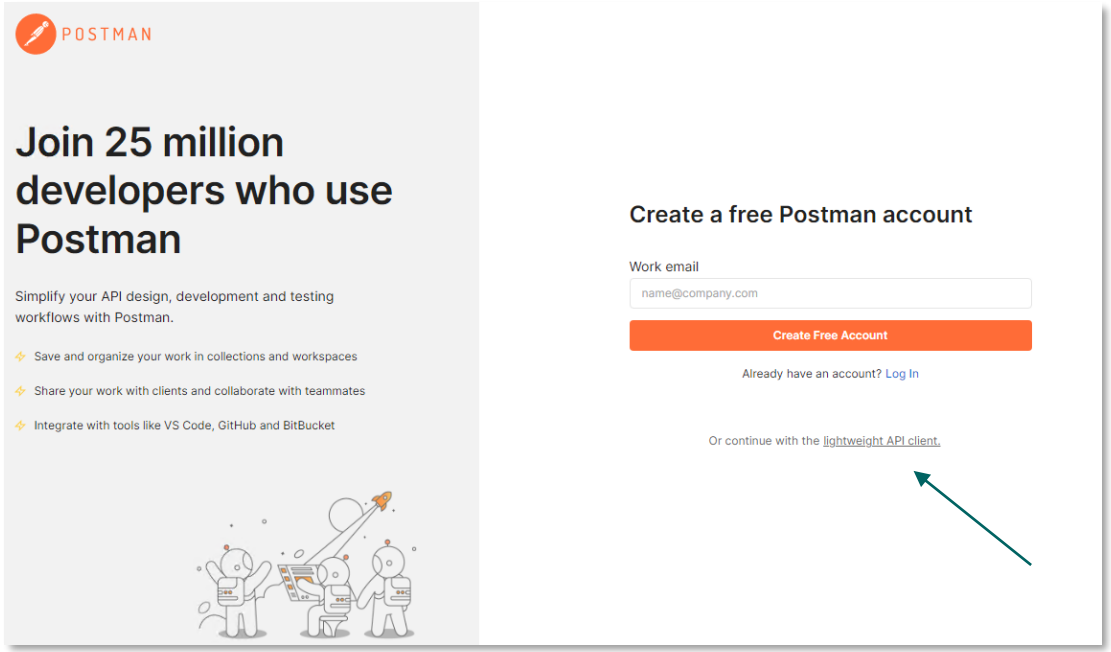
# Postman ® Installation

# Postman ® Installation



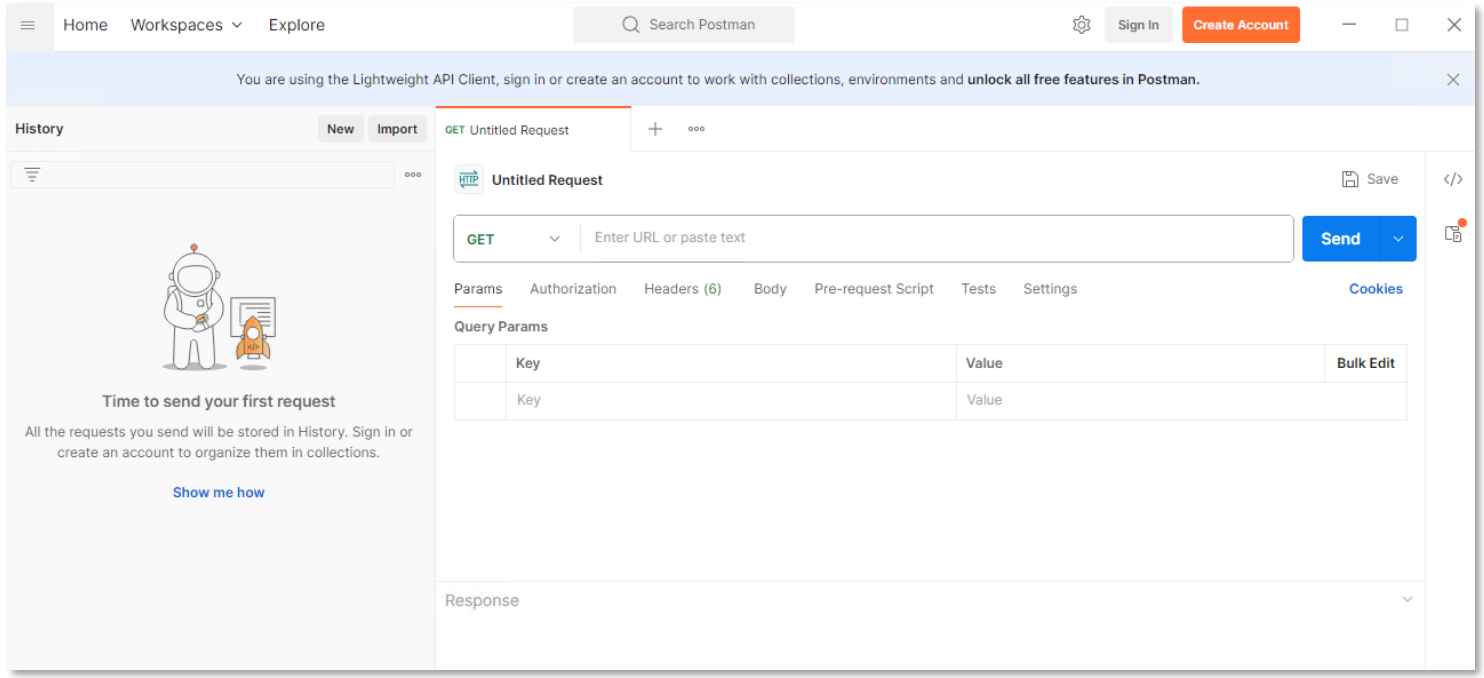

1. Visit the Postman website at https://www.postman.com/.
2. click the "Download for Windows" button once the download page opens.
3. Download The Executable




3. Postman offers two versions: 64-bit and 32-bit. Choose the version that matches your system architecture.
4. Click on the download button to start the download.
5. Once the installation is completed, you have the option of signing in or continuing with the lightweight API Client

# Postman ® Installation



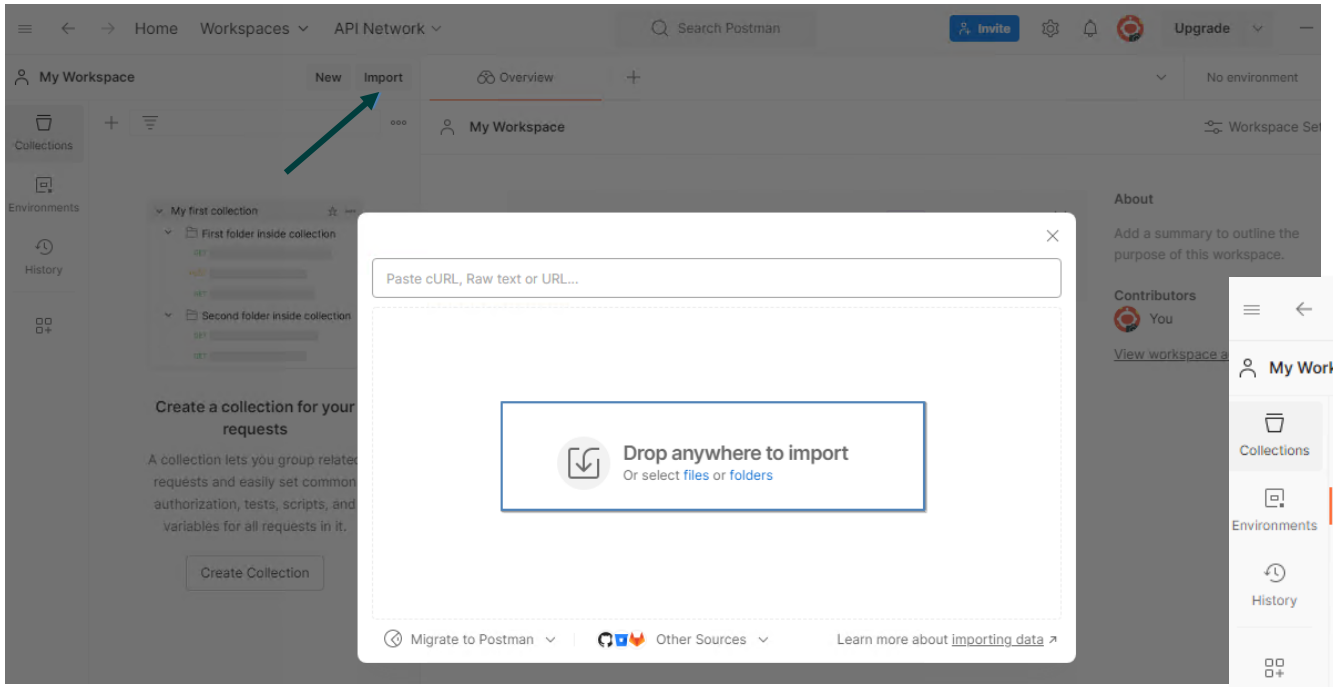

1. Once the installation is complete, you can launch Postman by finding it in your Start menu or by double-clicking the desktop shortcut if you choose to create one during installation.

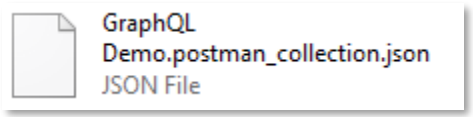
2. Upon launching Postman, you may be prompted to sign in or create a Postman account. This step is optional but provides access to additional features, such as syncing your collections across devices and importing collections

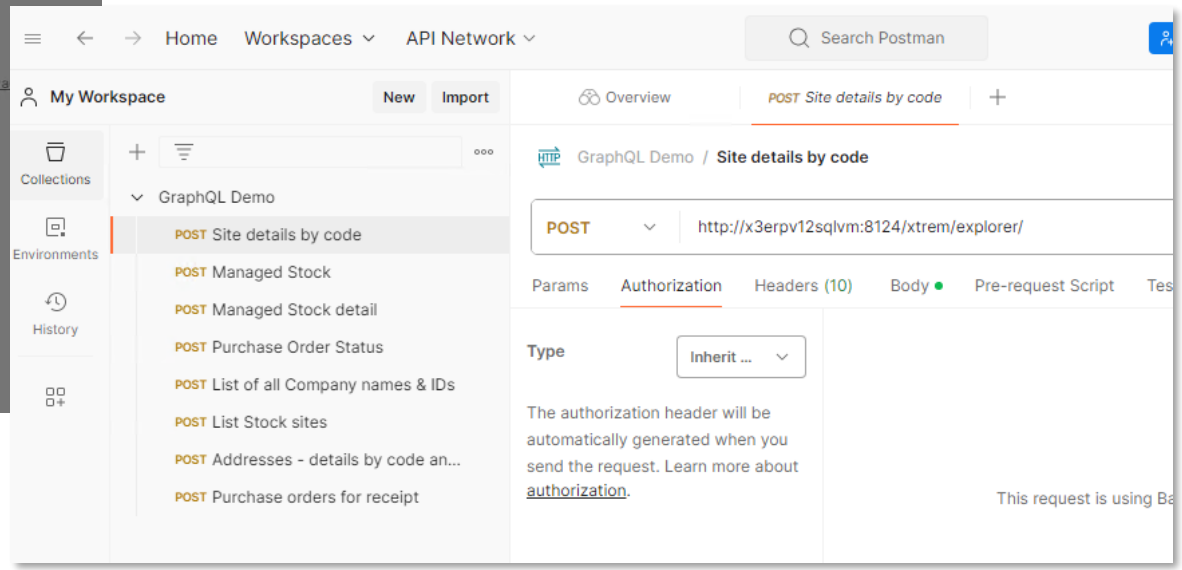
# Postman ® Sending Requests

# Importing Collections

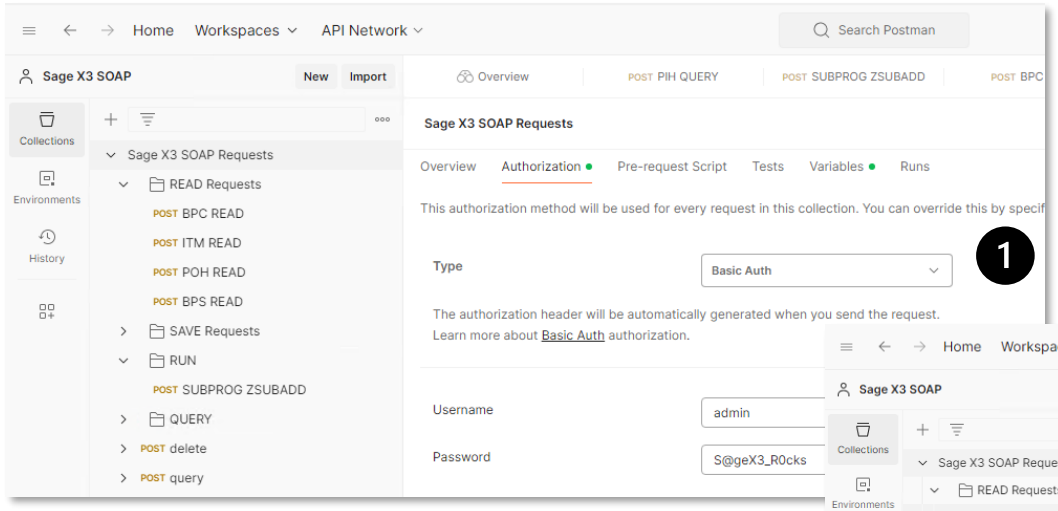
With the lightweight client, you are unable to import any existing collections; once you have created an account, you can import collections. Other advantages include syncing your collections and sharing with other approved users.



# Sending SOAP Request To X3

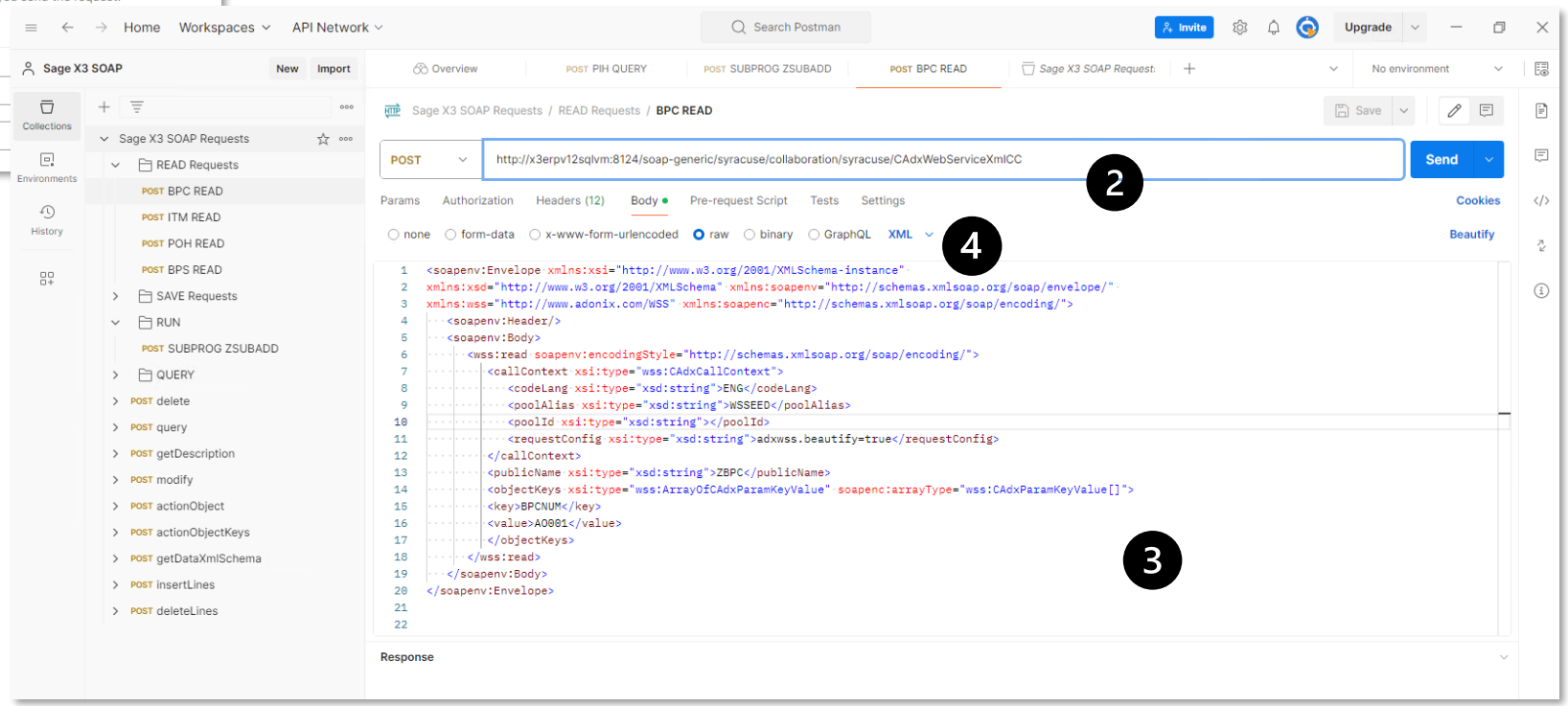



1. Configure the authentication this can be done at the individual request level or at the workspace level, and all requests can be set to inherit the authentication settings

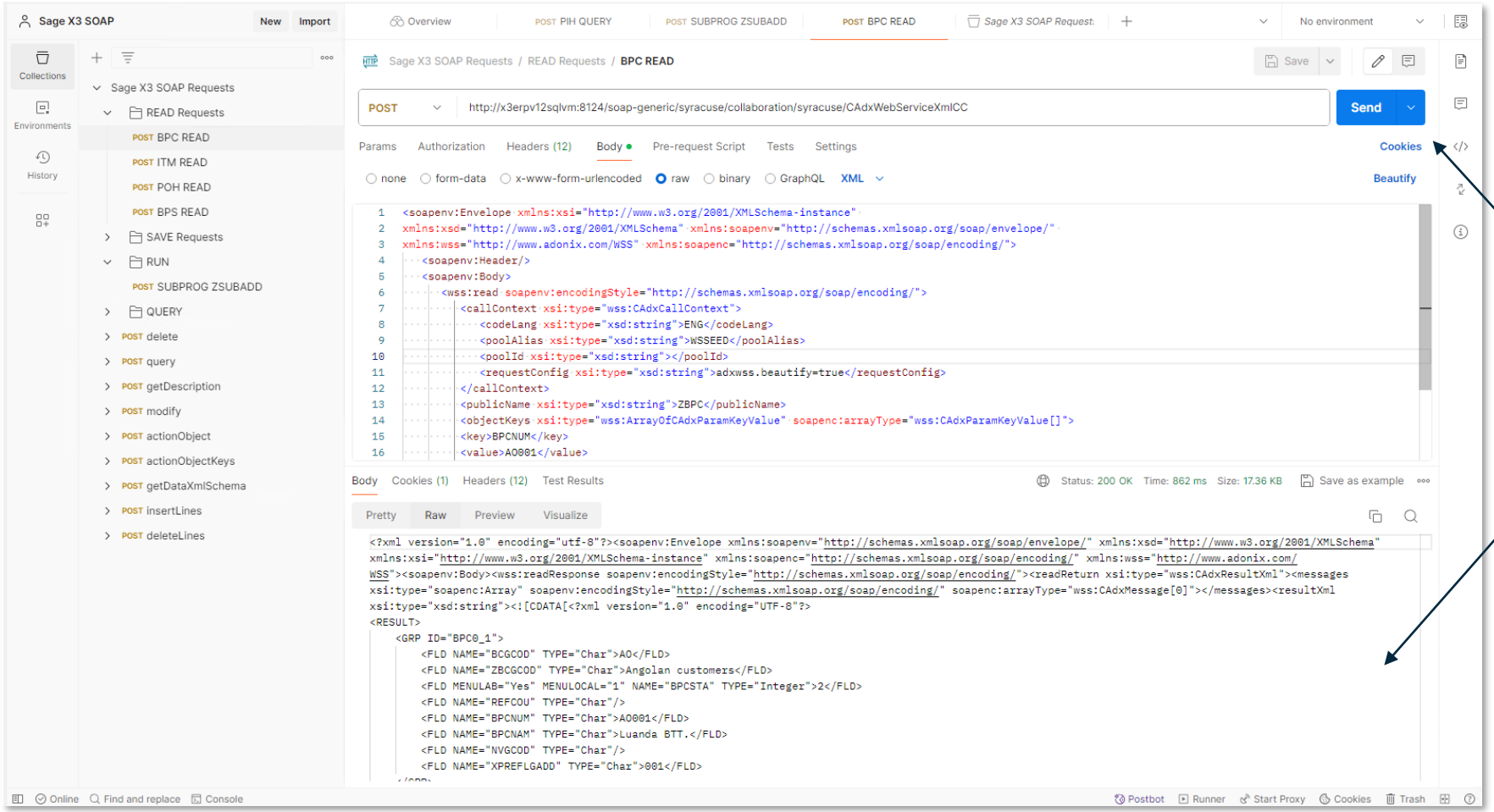
2. Populate the address of the web services WSDL and set the request type to POST

3. Populate the body of your SOAP request all parameters will be identical to those used in SoapUI

4. Set the request format in this case, raw XML



# Sending SOAP Request To X3




5. Press Send and the Response from the SOAP request is displayed

# Sage X3 and GraphQL

# Sage X3 and GraphQL

GraphQL is a query language for APIs and a runtime for executing those queries by a server. It was developed by Facebook in 2012 and later open-sourced in 2015. From Sage X3 2021 R3, we now have the possibility to use the GraphQL API to execute queries and modify Sage X3 data.

**Prerequisites**

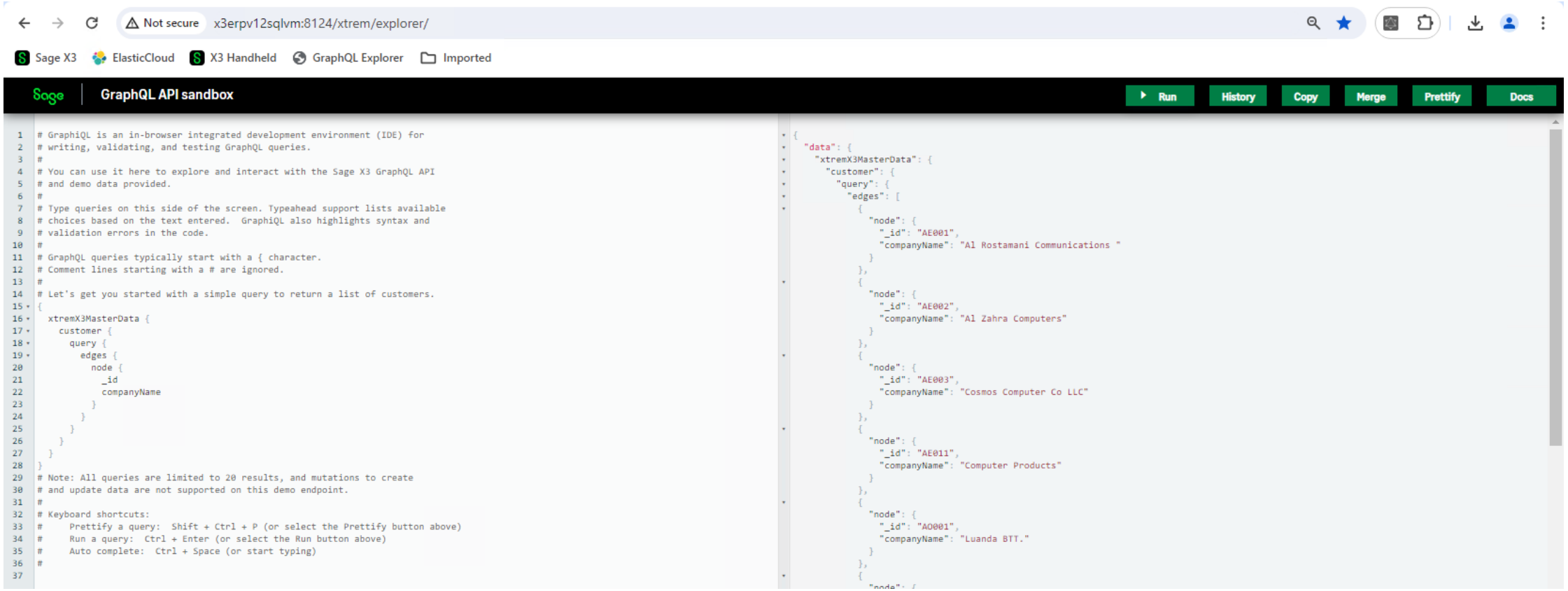
- Sage X3 services for GraphQL and ADC needs to be installed and configured to utilise the GraphQL API
    - Installing Services
    - Configuring the database connection in Syracuse
    - Detailed installation step & configuration steps - https://online-help.sagex3.com/erp/12/en-us/Content/V7DEV/getting-started_Sage-X3-Services-installation.html

- For ADC access, you need to have the ADC Badges

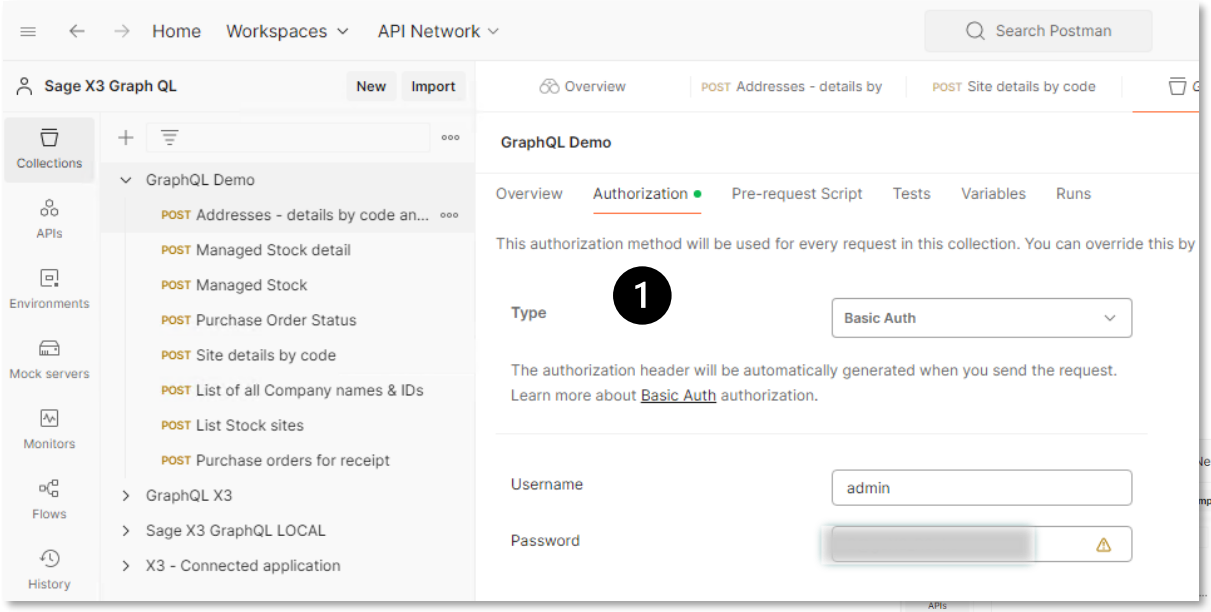
- Badges need to be assigned to the appropriate role that will be used by the ADC user

# Sage X3 and GraphQL

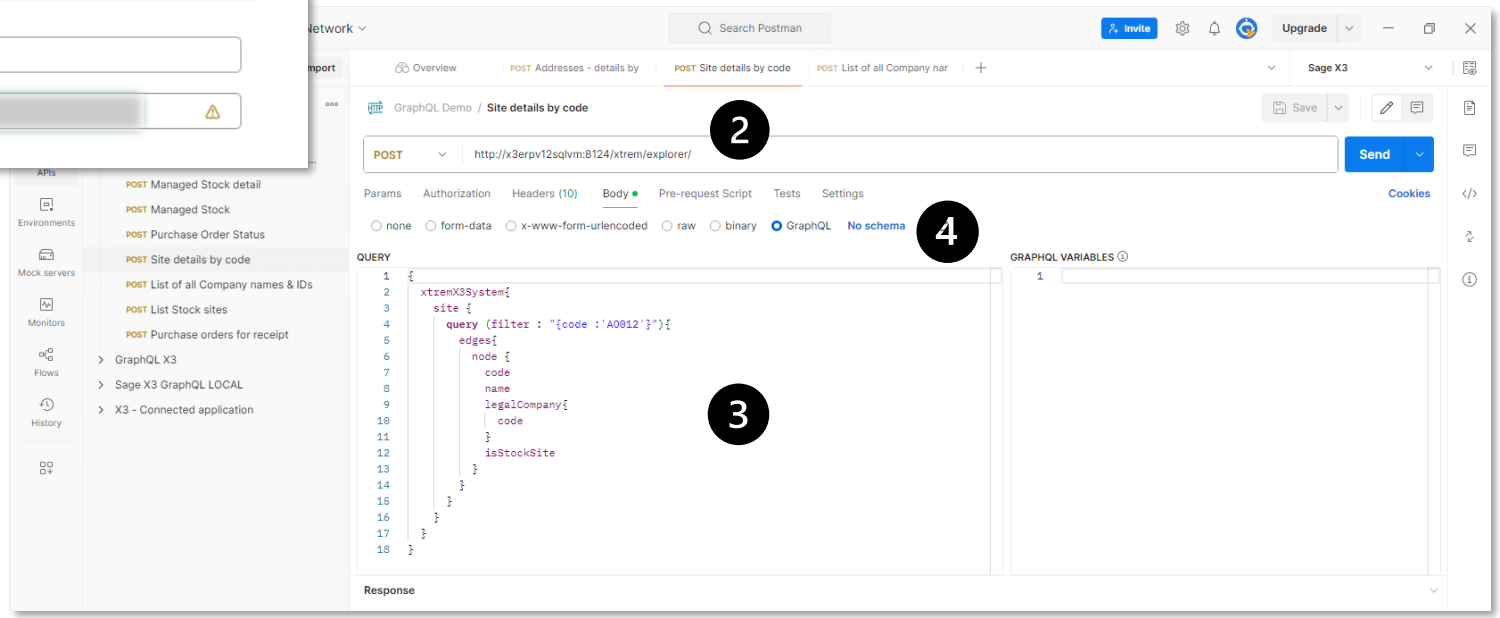
With Sage X3 Services installed, you can access the inbuilt GraphQL explorer, allowing you to interact with your Sage X3 data for the configured folder.



# Sending GraphQL Request To X3




1. Configure the authentication this can be done at the individual request level or at the workspace level, and all requests can be set to inherit the authentication settings

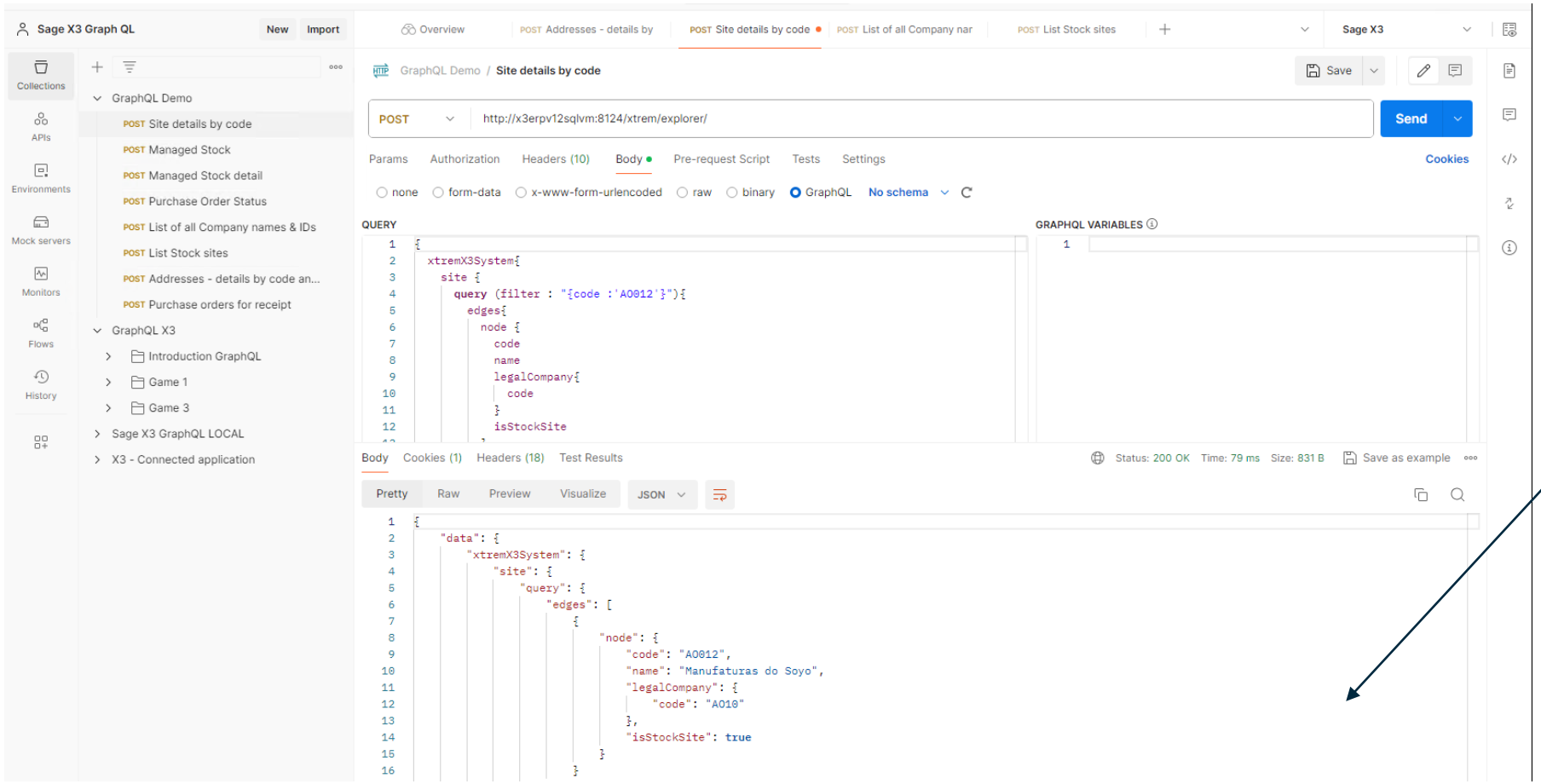



2. Populate the address of the GraphQL service endpoint

3. Populate the body of your GraphQL request

4. Set the request format in this case GraphQL

# Sending GraphQL Request To X3




5. Press Send and query results are displayed

# Demo

# Demo

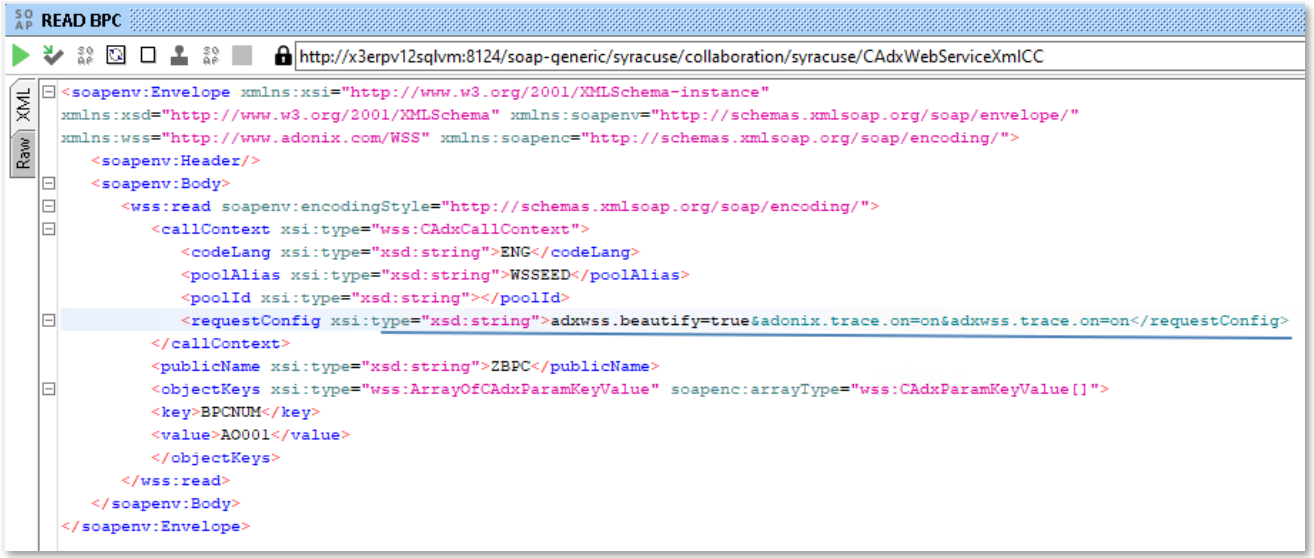
# Tips

# SOAP Request

## Request Config property

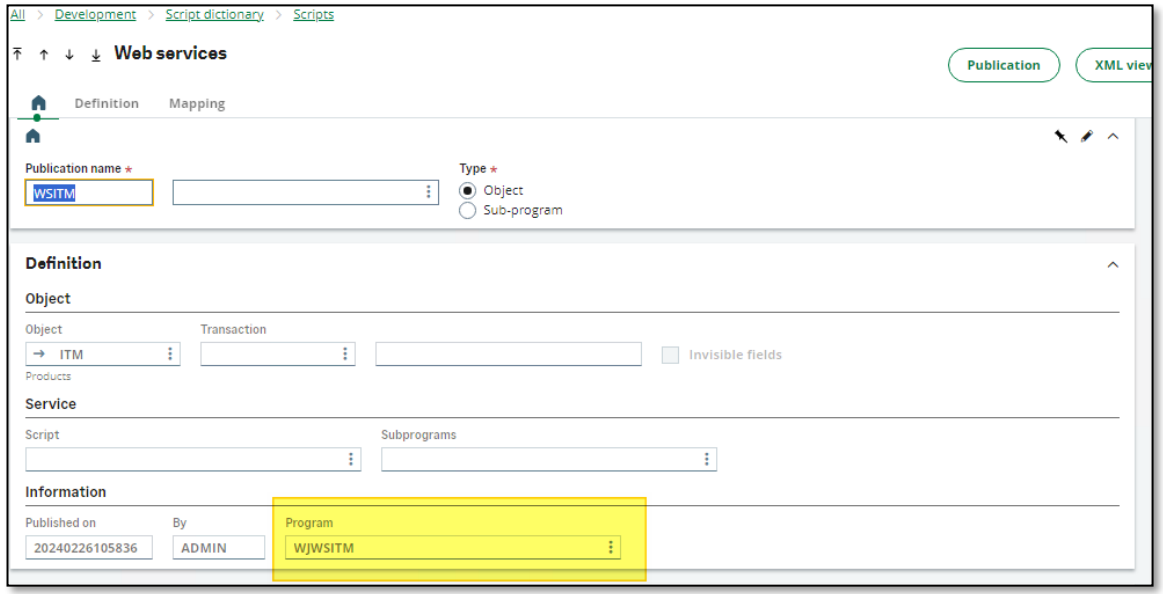
In the request config property, we can specify different settings separated by the '&' character to activate processing settings, which can be used for troubleshooting using the response to the web service call.

adxwss.trace.on=[on/off] – Activates web server (Syracuse layer) trace and embeds the trace in the response
adonix.trace.on=[on/off] – X3 (classic layer) server trace
adonix.trace.level=(1/2/3) - Specify server trace levels
adxwss.beautify=true – format the response
adxwss.optreturn=(XML/JSON) – Response Format

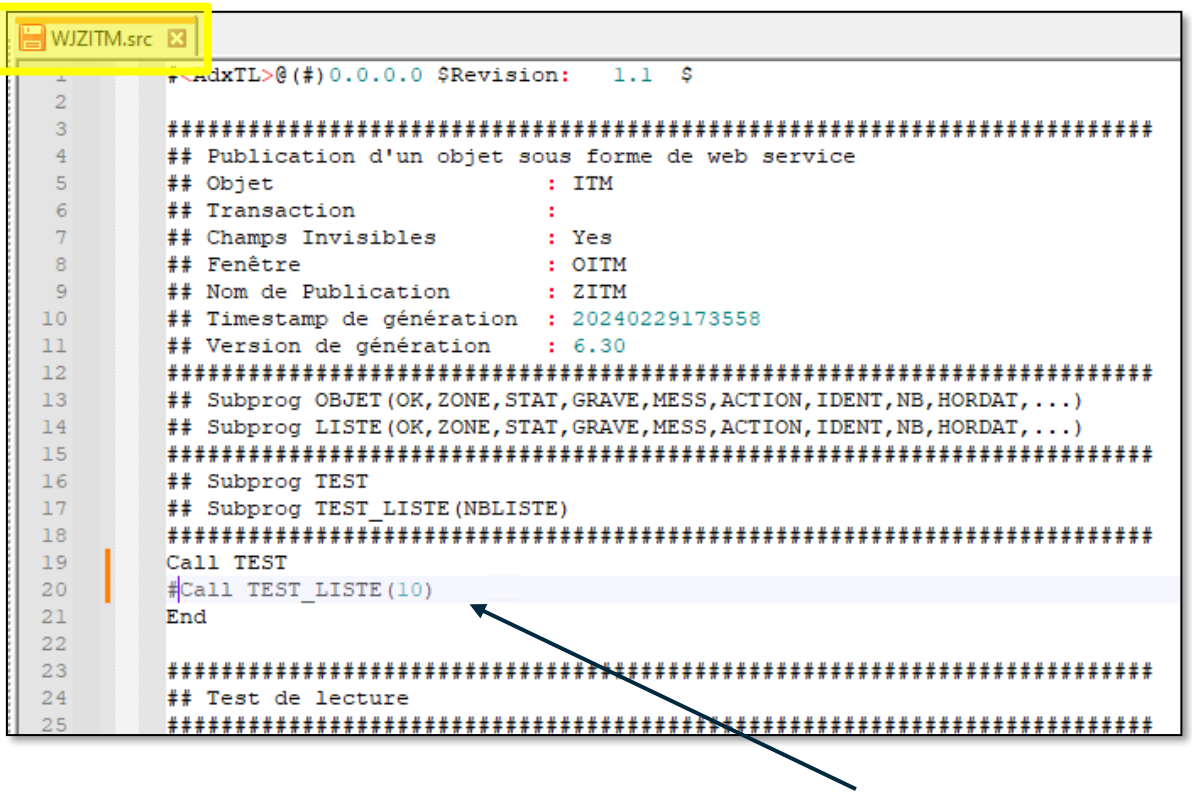


# When there is no SOAP Response

In a situation where your SOAP request is returning no response or not the expected response, we can trace the 4GL program the web service uses.




Open the file which will be in the folder TRT folder, alternatively you could modify using the script editor for example "D:\Sage\X3V12\folders\SEED\TRT\WJZITM.src"




Comment out #Call TEST_LISTE(10) enable Call TEST

# When there is no SOAP Response

Add the variable being passed in the request; this particular request was a READ on the ITMREF for the value BIKE, so in this case, we can add value BIKE to WW_IDENT in the web service program source.

WJZITM.src

```
10      ## Timestamp de génération  : 20240229173558
11      ## Version de génération    : 6.30
12      ################################################################################
13      ## Subprog OBJET(OK,ZONE,STAT,GRAVE,MESS,ACTION,IDENT,NB,HORDAT,...)
14      ## Subprog LISTE(OK,ZONE,STAT,GRAVE,MESS,ACTION,IDENT,NB,HORDAT,...)
15      ################################################################################
16      ## Subprog TEST
17      ## Subprog TEST_LISTE(NBLISTE)
18      ################################################################################
19      Call TEST
20      #Call TEST_LISTE(10)
21      End
22
23      ################################################################################
24      ## Test de lecture
25      ################################################################################
26      Subprog TEST
27      Local Char    TEXTE(20)
28      Local Integer I , J , K
29      Call OUVRE_TRACE("") From LECFIC
30      Gosub DEFVAR
31      WW_ACTION = "READ"      : # READ, CREATE, MODIFY, DELETE, SUPLIG, INSLIG
32      WW_IDENT  = "BIKE"        : # Clé de l Objet Val1~Val2
33
```

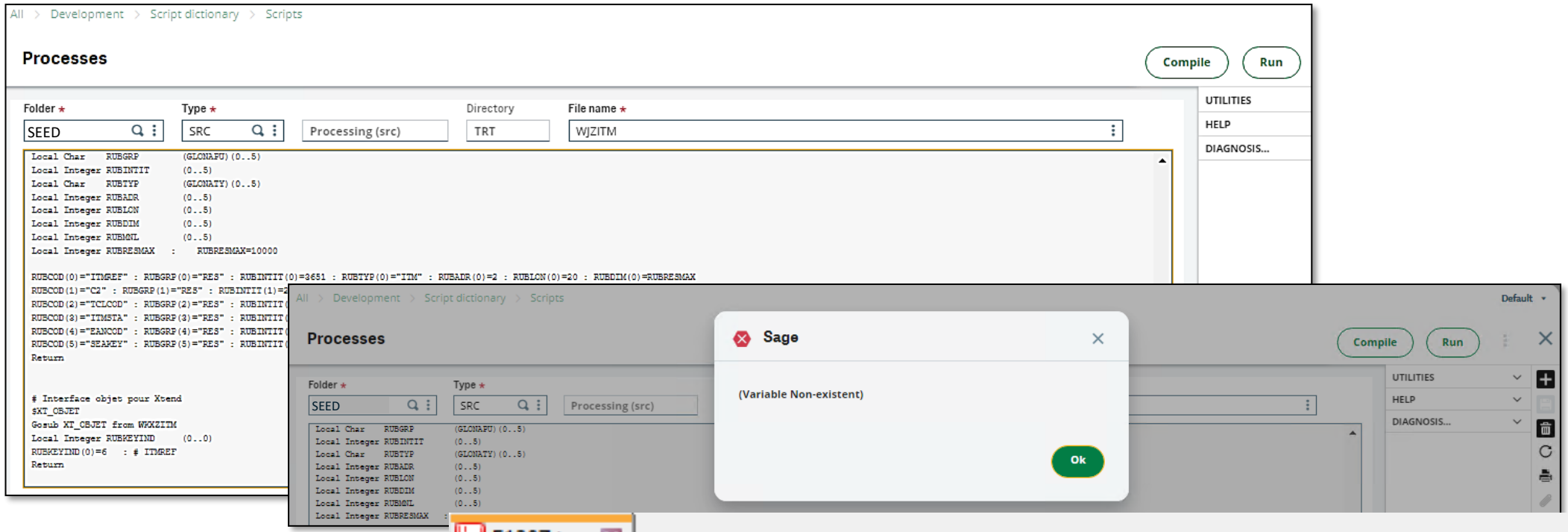
WJZITM.src

```
25      ################################################################################
26      Subprog TEST
27      Local Char    TEXTE(20)
28      Local Integer I , J , K
29      Call OUVRE_TRACE("") From LECFIC
30      Gosub DEFVAR
31      WW_ACTION = "READ"      : # READ, CREATE, MODIFY, DELETE, SUPLIG, INSLIG
32      WW_IDENT  = "BIKE"        : # Clé de l Objet Val1~Val2
33
34      Onerrgo ERREURWS
35      Gosub INITMAJ From ZZWSITM
36      Onerrgo
37
38      Gosub WEBSERV
39      Gosub ANALRESOBJ
40
41      Onerrgo ERREURWS
42      Gosub RESMAJ From ZZWSITM
43      Onerrgo
44
45      #Gosub DETAILRES     : # Pour afficher le détail des variables lues ou écrites
46      Call FERME_TRACE From LECFIC
47      Call LEC_TRACE From LECFIC
48      End
49
```

Enable/uncomment #Gosub DETAILRES function

# When there is no SOAP Response

In Sage X3 Script editor, RUN the web service program, and an error will be displayed on the screen

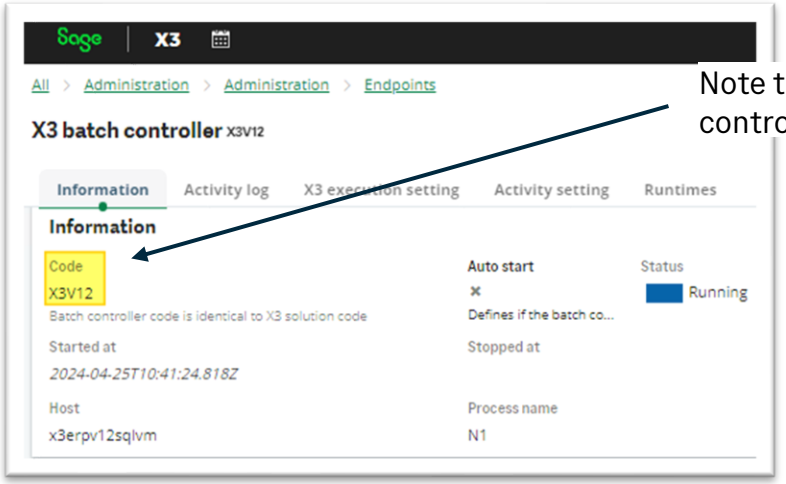



The full error is logged in a trace file in the TRA director of the folder , On this case a variable was missing

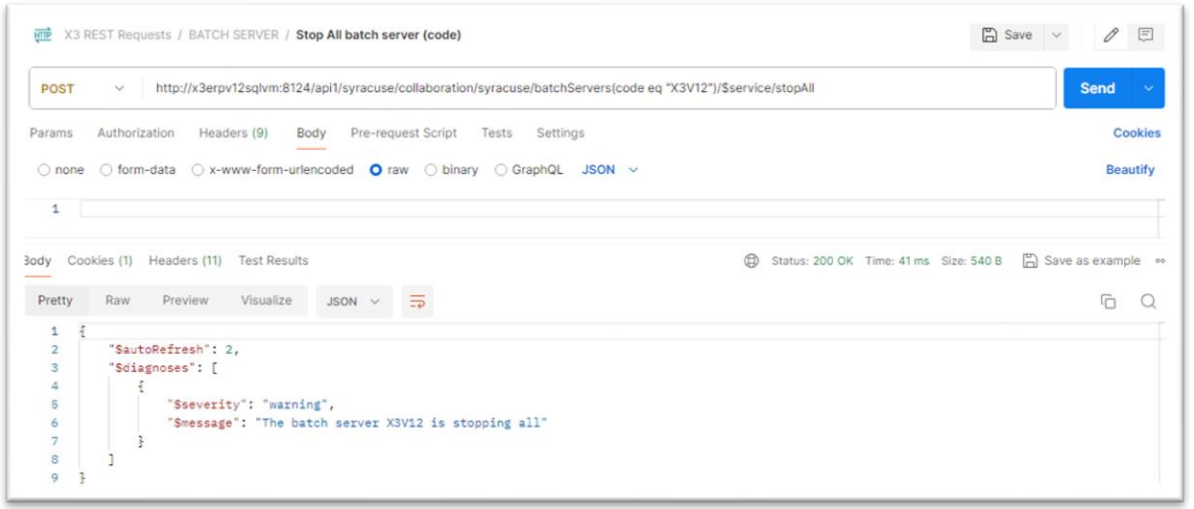
# Control Batch Server using Postman

Postman can also be used for REST requests , which can be used to control the Batch Server

**To stop all Post**
http://localhost:8124/api1/syracuse/collaboration/syracuse/batchServers(code eq "X3V12")/$service/stopAll
**To start Post**
http://localhost:8124/api1/syracuse/collaboration/syracuse/batchServers(code eq "X3V12")/$service/start




Note that X3V12 is the Batch controller code




Other tasks can also be performed, like getting the status of the batch server and the configuration information.

More info can be found in the online help https://online-help.sageerpx3.com/erp/12/staticpost/how-to-control-the-batcher-server/

# GraphQL Network Inspector

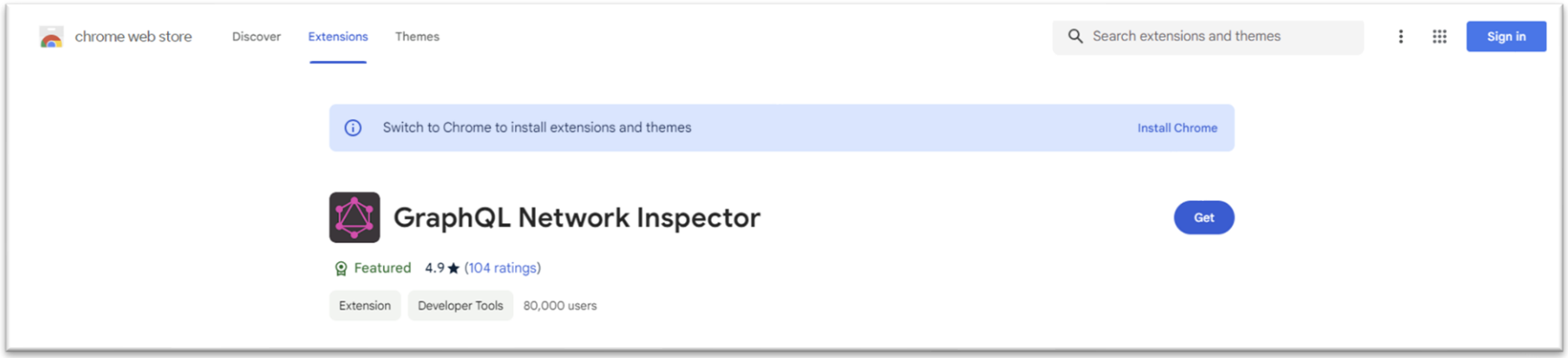
The GraphQL Network Inspector for Chrome is a tool designed to help debug and analyse GraphQL requests and responses in web applications.

With this extension, you can

View GraphQL Requests and Responses: It allows you to see the GraphQL requests made by the application and the corresponding responses returned by the server.

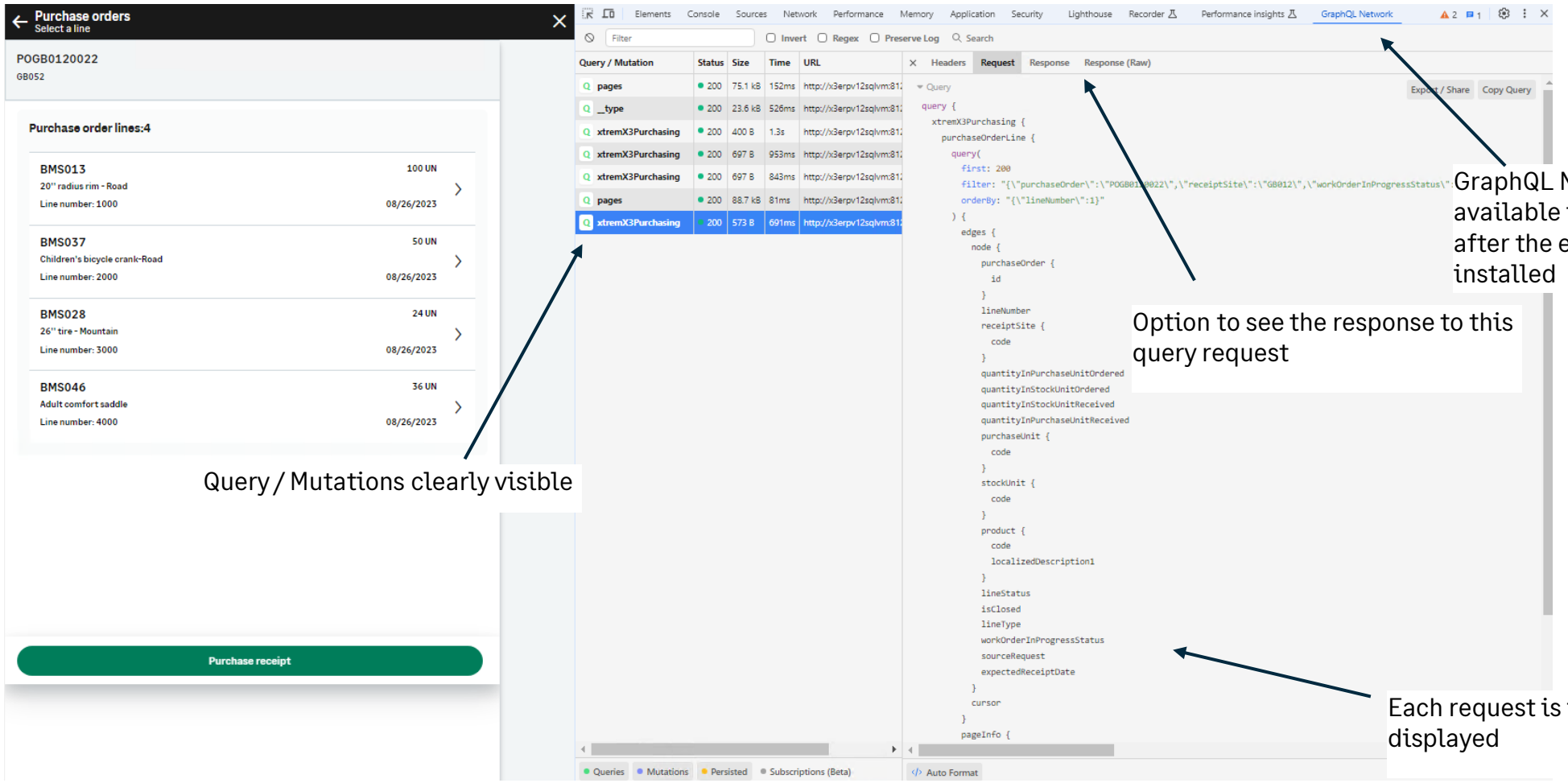
Extension can be used to debug GraphQL queries, identify errors, and optimize query performance.

Provides visibility and insights into the communication between their web applications and GraphQL servers.




GraphQL Network Inspector (google.com)

# GraphQL Network Inspector




Query / Mutations clearly visible

Option to see the response to this query request

GraphQL Network tab is available in developer tools after the extension has been installed

Each request is formatted and displayed

# Appendices

# Useful Links

| Resource | Link |
| --- | --- |
| Postman help | Postman \| Sage Developer |
| SoapUI® Official Help | An Introduction to API Testing with SoapUI® |
| REST Webservices in detail | Sage X3 Technical Support Tips and Tricks (May 2023) |
| Installing X3 Services | Sage X3 Services installation |
| Web Services in detail | Sage X3 Technical Support Tips and Tricks (March 2022) |
| How to use REST to GET and Post using Postman | How to use REST Web services to Get and Post a record |
| Graph QL Network Inspector | GraphQL Network Inspector (google.com) |
| Sage X3 Developer Portal | Develop for X3 \| Sage Developer |

# Thank you!