

Cloud Access & REST API

Team Developer 7.4.x

15 03 2022

| Helmut Reimann | Lead Solution Consultant

Agenda

- REST API
 - JSON.APL
- Remote Access

REST API

SalJSON.apl

New SalJSON class library ([SalJSON.apl](#)) to support more functionalities relating JSON and REST API

- Functional class [SalJSON_Config](#)
 - Configuration

Code

- ◆ Functional Class: SalJSON_Config
 - ◇ Description: Configuration class for SalJSON.
 - ◇ Derived From
 - ◇ Class Variables
 - ◆ Instance Variables
 - ◆ Constructor/Destructor
 - ◆ Functions
 - ◆ Function: SetAuthentication
 - ◆ Function: Decrypt
 - ◆ Function: SetTimeout
 - ◆ Function: EnableProxy
 - ◆ Function: SetProxy

REST API

SalJSON.apl

New SalJSON class library ([SalJSON.apl](#)) to support more functionalities relating JSON and REST API

- Functional class **SalJSON**
 - Working with API

Code

- ◆ Functional Class: SalJSON
 - ◇ Description: SalJSON class to serialize/deserialize JSON string from/to UDV data, and support REST API.
 - ◇ Derived From
 - ◇ Class Variables
 - ◆ Instance Variables
 - ◆ Constructor/Destructor
 - ◆ Functions
 - ◆ Function: SetMethod
 - ◆ Function: SetRequestData
 - ◆ Function: Serialize
 - ◆ Function: Deserialize
 - ◆ Function: AddHTTPHeader
 - ◆ Function: ClearHttpHeaders
 - ◆ Function: AddUrlParameter
 - ◆ Function: ClearUrlParameters
 - ◆ Function: SendRequest

REST API

SalJSON.apl

Sample:

- Set Authentication
- Set new timeout value
- Check if set (for testing only...)
- Set method to HTTP_GET

Code

```
◇ Call oJsonConfig.SetAuthentication( 'hreimann', ' ' )  
◇ Set dfTimeOut = oJsonConfig.SetTimeout( nParSeconds )  
◇ Set dfTimeOutNew = oJsonConfig.SetTimeout( nParSeconds )  
◇ Call oJson.SetMethod( HTTP_GET )
```

REST API

REST API Sample

Get data from a REST WS

- „oRuntime“ = Instance of a functional class. This class contains all instance variables which are filled by the operation of the REST API:

```
◆ Functional Class: clsJASONCurrentRuntime
  ◇ Description:
  ◇ Derived From
  ◇ Class Variables
  ◆ Instance Variables
    ◇ String: runtime_version
    ◇ Boolean: enabled
    ◇ Boolean: deleted
    ◇ String: client_package_id
    ◇ String: client_platform
    ◇ String: proxy_package_id
    ◇ String: proxy_platform
    ◇ String: description
    ◇ String: proxyAPILevel
    ◇ Boolean: default
    ◇ String: installdate
```

Code

```
◆ Local variables
  ◇ clsJASONCurrentRuntime: oRuntime
```

```
◇ Set sREST_User = sgUser
◇ Set sREST_PWD = sgPWD
◇ Set sURL = "https://" || sgURL || sgAPIRuntime
◇ Set dfURL = sURL
◇ !!
◇ Call SaJSONDeserializeUDV( "oRuntime", sURL, sREST_User, sREST_PWD, sErr )
◇ Set dfError = fCleanUpErrorMessage( sErr )
◇ !
◇ Set dfVersion = oRuntime.runtime_version
◇ Set dfPackageID = oRuntime.client_package_id
◇ Set dfClientPlatform = oRuntime.client_platform
```

REST API

REST API Sample

Passing data into a REST WS

- „oUser“ is a complex structure with nested objects

◆ Functional Class: clsJASONNewUser

- ◇ Description: creates a new user
- ◇ Derived From
- ◇ Class Variables
- ◆ Instance Variables
 - ◇ String: login
 - ◇ String: name
 - ◇ String: email
 - ◇ ! String: role
 - ◇ ! Boolean: disabled
 - ◇ clsUserRole: role
- ◇ Constructor/Destructor

◆ Functional Class: clsUserRole

- ◇ Description:
- ◇ Derived From
- ◇ Class Variables
- ◆ Instance Variables
 - ◇ String: id
 - ◇ String: name
- ◇ Constructor/Destructor
- ◇ Functions

Code

◇ clsJASONNewUser: oUser

```
◇ Set oUser.name = dfNewName
◇ Set oUser.email = dfNewMail
◇ Set oUser.login = dfNewLogin
◇ Set oUser.role.name = cmbRole
◇ !
◇ Set sREST_User = sgUser
◇ Set sREST_PWD = sgPWD
◇ !
◇ Set sURL = "https://" || sgURL || sgAPIUsers
◇ !
◇ !
◇ Call SaJSONSerializeUDV("oUser",sURL, HTTP_POST,
                          sResult ,sREST_User, sREST_PWD,sErr)
◇ Set dfError = fCleanUpErrorMessage( sErr )
```

Demo

Remote Access

- Working from Home
- Cloud Access

Demo

OpenText Exceed TurboX benefits

What's different about Exceed TurboX?

Optimal Performance and Bandwidth Usage

Efficient in bandwidth usage via ThinX Protocol.

Accurate graphical rendering.

Central Administration & Management

Centrally manage users.
View session activities.
Easily deploy upgrades.

Extremely Stable & High Security

High availability.
Load balancing.
Strong encryption and data protection.

Mixed UNIX/Linux & Windows Environments

Enables remote apps for UNIX/Linux and Windows app servers.

Windows, Mac, Linux and web clients.

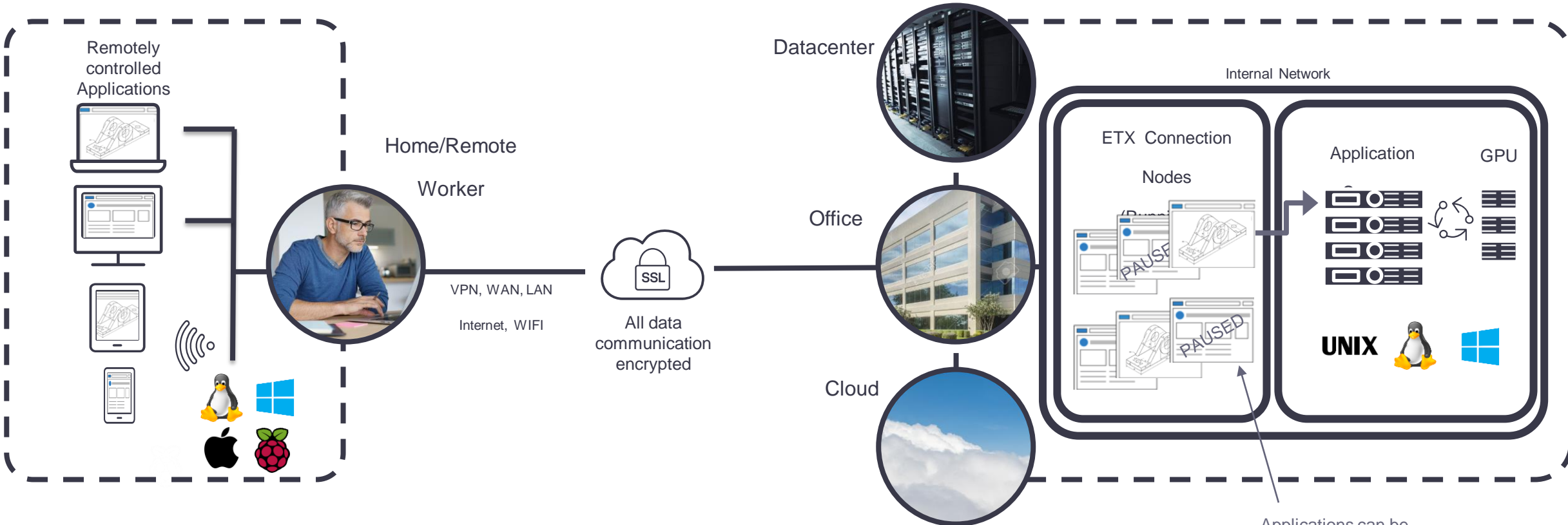
Sharing & Collaboration Features

Suspend and Resume.
Session Sneak Peek.
Supercharged Sharing.



Exceptional performance and control for remote apps and desktops

How it Works



Applications can be paused and resumed from any device.

Long-term Value and ROI



FOR CORPORATE



Lower Costs



Higher User Productivity



Secured IP



FOR END-USERS



Responsive Applications



Work from Anywhere



Better Collaboration



FOR IT



Centralized Administration



Secured Environment



Faster Response Times

VALUE ACROSS THE ORGANIZATION

Demo

opentext™

Q & A

opentext™

Thank you



twitter.com/opentext



linkedin.com/company/opentext

More information!

<https://bit.ly/Gupta-TD-free-trial>

<https://www.opentext.com/etx>

[**opentext.com/gupta**](https://www.opentext.com/gupta)