

Sample Service Suite Scripts

Here are some Service Suite samples scripts that can be used as a reference.

Basic Authentication

```
/**
 * btoa
 * @param st
 * @returns base64 encoded string
 */
function btoa(st) {
    return Java.type("java.util.Base64").getEncoder().encode(st);
}

var $username = "admin";
var $password = "Password@1";
var $TOKEN = "Basic "+btoa($username+": "+$password);

addGlobalHeader("authorization", $TOKEN);

var url = "https://someServer.com/api/somecall";
var data = {"short_description":"some data",
    "description":"some description"};

var call = createPost(url, data);
call.addHeader("Content-Type", "application/json");

var resp = process(call);
```

Demosite

```

var formPage = get("https://demosite.appvance.com/products/ruby-on-rails-tote");
log(getCookies("https://demosite.appvance.com/"));
var token = extract("html", formPage,
'//form[@action="/orders/populate"]/input[@name="authenticity_token"]/@value');
log("Extracted token: "+token);

var form = formData();
form.put("authenticity_token", token);
form.put("utf-8", decodeURIComponent("%E2%9C%93"));
form.put("variant_id", "1");
form.put("quantity", "1");
form.put("button", "");

var postData = createPost("https://demosite.appvance.com/orders/populate", form);
var purchaseResults = process(postData);

log("response:\n"+purchaseResults.getDataString());

assertContains(get("https://demosite.appvance.com/cart?variant_id=1").getDataString(), "Ruby on
Rails Tote");
assertContains(get("https://demosite.appvance.com/cart_link").getDataString(), "$15.99");

```

Login to AIQ

```

$baseUrl = "http://localhost:8080";
var $loginForm = formData();

$loginForm.put("username", "appvance");
$loginForm.put("password", "appvance");

var $call = createPost($baseUrl+"/AppvanceServer/rest/admin/loggin", $loginForm);
var $resp = process($call);
var $json = JSON.parse($resp.getDataString());

if ($json.logged) {
    var $jsession = getCookieValue($baseUrl, "JSESSIONID");
    addGlobalHeader("cookie", "JSESSIONID="+$jsession); // adding global cookie for JSESSIONID
    log("Logged In, JSESSIONID: "+$jsession);
}

```

```
} else {  
    log("Unsuccessful login");  
}
```

ServiceSuite and XML

```
var payload = "<xml><username>John</username><id>123-1341-123</id></xml>";  
var call = createPost("https://example.com/getAddress", payload, "text/xml");  
var result = process(call);  
var address1 = extract("XML", result, "//Address1");  
var city = extract("XML", result, "//City");  
var state = extract("XML", result, "//ST");
```

Upload a file to AIQ repository

```
const $boundary = "AIQ-DS-BOUNDARY";  
  
function login($baseUrl, $username, $password) {  
    var $loginForm = formData();  
    $loginForm.put("username", $username);  
    $loginForm.put("password", $password);  
    var $call = createPost($baseUrl + "/AppvanceServer/rest/admin/loginForServices",  
$loginForm);  
    var $resp = process($call).getDataString();  
    if ($resp.indexOf("true") != -1) {  
        var $jsession = getCookieValue($baseUrl, "JSESSIONID");  
        addGlobalHeader("cookie", "JSESSION=" + $jsession);  
        log("Login successful")  
        return true;  
    } else {  
        log("Login was NOT successful")  
        return false;  
    }  
}
```

```

function createMultipartPayload($filename) {
    var header = '--' + $boundary + '\n' +
        'Content-Disposition: form-data; name="file"; filename="' + $filename + '"\r\n' +
        'Content-Type: application/octet-stream\r\n\r\n';
    var headerBytes = new Array(header.length);
    for (var i = 0; i < header.length; ++i) {
        headerBytes[i] = header.charCodeAt(i);
    }
    var docBytes = readBytes($filename);
    var footer = '\r\n--' + $boundary + "--\r\n";
    var footerBytes = new Array(footer.length);
    for (var i = 0; i < footer.length; ++i) {
        footerBytes[i] = footer.charCodeAt(i);
    }
    return headerBytes.concat(docBytes).concat(footerBytes);
}

var $filename = "test2.js";    // CHANGE FOR YOUR FILE TO UPLOAD
var $repositoryName = "test"; // CHANGE FOR YOUR REPO NAME, THIS CONF MUST BE IN THE USER
                              ACCOUNT USED TO LOGIN
var $username = "appvance";
var $password = "appvance";

login("http://my-aiq-controller.appvance.com:8080", $username, $password); // CHANGE FOR
YOUR AIQ SERVER DNS

var data = createMultipartPayload($filename);
log(data.length, $username, $password);

var call= createPost(
    "http://localhost:8080/AppvanceServer/rest/file/uploadFile?&commitMessage=Appvance+skip+commit
    &path=REPOSITORY/" + $repositoryName + "/" + $filename, data, "multipart/form-data", " boundary=" +
    $boundary);

var resp = process(call);
log(resp.getDataString())

```

Upload the File to an application

```
/**
 * Creates a multipart payload from file contents
 */
function createMultipartPayload($filename) {
    var header = '--' + $boundary + '\n' +
        'Content-Disposition: form-data; name="file"; filename="' + $filename + '"\r\n' +
        'Content-Type: application/octet-stream\r\n\r\n';
    var headerBytes = new Array(header.length);
    for (var i = 0; i < header.length; ++i) {
        headerBytes[i] = header.charCodeAt(i);
    }
    var docBytes = readBytes($filename);
    var footer = '\r\n--' + $boundary + "--\r\n";
    var footerBytes = new Array(footer.length);
    for (var i = 0; i < footer.length; ++i) {
        footerBytes[i] = footer.charCodeAt(i);
    }
    return headerBytes.concat(docBytes).concat(footerBytes);
}

/**
 * Uploads a file to $SERVER
 */
function uploadFile($filename) {
    var data = createMultipartPayload($filename);
    var url = $SERVER+ "/upload/" + $filename + "?filename=" + $filename;
    log("url: " + url);
    var call= createPut(url, data, "multipart/form-data", " boundary=" + $boundary);
    return process(call).getDataString();
}

const $boundary = "AIQ-DS-BOUNDARY";
const $SERVER = "http://xxxx.eu-west-2.elb.amazonaws.com";

var $filename = "test2.js";    // CHANGE FOR YOUR FILE TO UPLOAD
var result = uploadFile($filename);
```

```
log(result);
```

Send an email with a custom timestamp in the Subject

```
var date = new Date();
var formatDate = date.getDate() + "-" + date.getMonth() + "-" + date.getFullYear() + " " +
date.toLocaleTimeString();
var graphService = createOAuthMailClient("email", "clientId", "secretKey", "tenantID");
var msg = graphService.createMailMessage("Message generated " + formatDate, "Test body email."
);
msg.addTo(["to@test.com"]);
var sent = graphService.sendMessage(msg);
if (!sent) {
    throw new Error('An error occurred while sending the email.');
```

```
} else {
    log("Email sent.");
}
```

Extract value from receive email's subject

```
var graphService = createOAuthMailClient("email", "clientId", "secretKey", "tenantID");
var unreadList = graphService.getUnreadMessages();
var unreadCount = unreadList.size();
if (unreadCount > 0) {
    var i = 0; //first unread email
    log("Id: " + unreadList.get(i).id);
    log("From: " + unreadList.get(i).from.emailAddress.address);
    log("Subject: " + unreadList.get(i).subject);

    var subject = unreadList.get(i).subject;
    var startIndex = subject.lastIndexOf("Ref. ") + 5;
    var refCode = subject.substr(startIndex, 8);
    log("Ref code: " + refCode);
```

Compare email's body

```
var expectedBody = "This is an automated acknowledgement to let you know that your  
correspondence to the Appvance Information Rights Team has been received.Any request for  
personal data will be handled as a data subject access request.";

var body = unreadList.get(i).body.content.replace(/(\r\n|\n|\r)/gm, ""); //remove line break  
and spaces.

body = body.substr(0,body.indexOf("uk")+2); //trim to remove links at the end
log("Email have the expected content: " + (body === expectedBody));
} else {
    log("There's no unread messages.");
}
```

Utility Functions

```
/**
 * btoa
 * @param st
 * @returns base64 encoded string
 */
function btoa(st) {
    return Java.type(java.util.Base64).getEncoder().encode(st);
}

/**
 * atob
 * @param st
 * @returns decoded base64 string
 */
function atob(st) {
    return Java.type(java.util.Base64).getDecoder().decode(st);
}
/**
```

```

*
* @param length number
* @returns random number of length number of digits
*/
function randomNumber(length) {
    var result = '';
    for (var i = 0; i < length; i++) {
        result += Math.floor(Math.random() * 10);
    }
    return result;
}

/**
 * Extracts catching an exception and returning NOT_FOUND string if target is not found.
 * @param {string} format ("HTML", "HTTP", "PDF", "CSV", "Excel", "TEXT", "STRING")
 * @param {HttpResponse} response
 * @param {string} target
 * @returns
 */
function extractOrNotFound(format, response, target) {
    const NOT_FOUND = "Not found.";
    var value;
    try {
        value = extract(format, response, target);
        if (typeof value == "undefined") {
            return NOT_FOUND;
        }
        return value;
    } catch (e) {
        return NOT_FOUND;
    }
}

/**
 * SSN
 * @returns random string matching social security number format
 */
function SSN() {
    var result = '';

```

```

    result = randomNumber(3) + "-" + randomNumber(2) + "-" + randomNumber(4);
    return result;
}

// *****
// parseCurrencyStringToNumber
//
// Parse a currency string to a number, $1,245.50 -> 1245.5
// *****

function parseCurrencyStringToNumber(currency) {
    if (typeof currency == "undefined" || currency == "") {
        log("Warning in parseCurrencyStringToNumber, expecting a number but got undefined or empty.");
        return -1;
    }
    if (currency == "0") {
        return 0;
    }
    if (currency.indexOf("(") !== -1) {
        currency = "-" + currency.replace(/[()]/g, "");
    }
    return Number(String(currency).replace(/^[^0-9.-]+/g, ""));
}

// *****
// parseNumberToCurrency
//
// Parse a number to a string currency with commas, 1245.5 -> $1,245.50
// *****

function parseNumberToCurrency(aNumber) {
    if (aNumber == 0) {
        return "$0.00";
    };
    var tempST = String(aNumber).replace(/^[^0-9.-]+/g, "");
    try {
        var tempNumber = Number(parseFloat(tempST).toFixed(2));
        var tempLocale = tempNumber.toLocaleString(undefined, { minimumFractionDigits: 2 });
        return "$" + tempLocale;
    } catch (e) {
        log("Error converting to currency [" + aNumber + "], isNaN: " + isNaN(aNumber) + " - "

```

```

+ e.message);
    return "NA";
}
}

/**
 * dateToString
 * @param day
 * @returns
 */
function dateToString(day) {
    var dd = String(day.getDate()).padStart(2, '0');
    var mm = String(day.getMonth() + 1).padStart(2, '0'); //January is 0!
    var yyyy = day.getFullYear();
    return mm + '/' + dd + '/' + yyyy;;
}

function getTodayDate() {
    var now = java.time.LocalDateTime.now();
    var formatter = java.time.format.DateTimeFormatter.ofPattern("MM/dd/yyyy");
    return now.withHour(0).withMinute(0).withSecond(0).format(formatter);
}

function getYesterdayDate() {
    var now = java.time.LocalDateTime.now();
    var formatter = java.time.format.DateTimeFormatter.ofPattern("MM/dd/yyyy");
    return now.minus(1, java.time.temporal.ChronoUnit.DAYS).withHour(0).withMinute(0)
        .withSecond(0).format(formatter);
}

function getTomorrowDate() {
    var now = java.time.LocalDateTime.now();
    var formatter = java.time.format.DateTimeFormatter.ofPattern("MM/dd/yyyy");
    return now.plus(1, java.time.temporal.ChronoUnit.DAYS).withHour(0).withMinute(0)
        .withSecond(0).format(formatter);
}

/**
 * getDateISOString
 * @returns

```

```

*/
function getDateISOString() {
    var st = new Date().toISOString();
    return st.replace(/-/g, "_").replace(/:/g, "_").replace(/\./g, "_");
}

/**
 * dateOneMonthAfter
 * @param $date
 * @returns
 */
function dateOneMonthAfter($date) {
    var day = new Date(Date.parse($date));
    day.setMonth(day.getMonth() + 1);
    day.setTime(day.getTime() + TIME_OFFSET);
    return dateToString(day);
}

/**
 * dateOneYearAfter
 * @param $date
 * @returns
 */
function dateOneYearAfter($date) {
    var day = new Date(Date.parse($date));
    day.setFullYear(day.getFullYear() + 1);
    day.setTime(day.getTime() + TIME_OFFSET);
    return dateToString(day);
}

/**
 * randomLastName
 * @returns
 */
function randomLastName() {
    var syllabs = 2 + Math.floor(Math.random() * 2)
    var result = '';
    var characters = 'bccdfghjklmnprrssttvwy';
    var vocals = 'aeioeua';
    var charactersLength = characters.length;

```

```
result += "SBCLKJMCH".charAt(Math.floor(Math.random() * 9));
result += vocals.charAt(Math.floor(Math.random() * 7));
syllabs--;
for (var i = 0; i < syllabs; i++) {
    result += characters.charAt(Math.floor(Math.random() * charactersLength));
    result += vocals.charAt(Math.floor(Math.random() * 7));
}
result += characters.charAt(Math.floor(Math.random() * charactersLength));
return result;
}
```

Last updated: June 02, 2023

If you have any feedback on this topic, please contact us at documentation@appvance.ai

Revision #2

Created 2026-07-31 15:39:35 UTC by Admin

Updated 2026-07-31 15:54:06 UTC by Admin