|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Retrieve security token | | |
| **URL** | api/tokens | | |
| **Http Method** | GET | | |
| **Parameter** | **Name** | **Type** | **Remark** |
| *userLogin* | *string* | *Mandatory* |
| *password* | *string* | *Mandatory - need to encrypt with AES algorithm, 128 bits key length, 128 bits block size, CBC (Cipher Block Chaining) operation modus, PKCS#7 padding*  *Key = YGjDImfP5bf8F7kGbdqWcA==*  *IV = bDA11yRcbUsTC9jl5LnuiQ==* |
| **Return** | Single object with the following property | | |
| **Name** | **Type** | **Remark** |
| *token* | *string* |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Retrieve WIP | | |
| **URL** | api/wips | | |
| **Http Method** | GET | | |
| **Parameter** | **Name** | **Type** | **Remark** |
| *serialNumber* | *string* | *Mandatory* |
| *customerName* | *string* | *Optional* |
| *materialName* | *string* | *Optional* |
| *materialPlantCode* | *string* | *Optional* |
| **Return** | Single object with the following property | | |
| **Property Name** | **Type** | **Remark** |
| *wips* | *wip* | *Collection of objects* |
|  | | |
| wip | | |
| **Property Name** | **Type** | **Remark** |
| *bomAlternativeBom* | *string* |  |
| *bomHeaderRev* | *string* |  |
| *bomId* | *string* |  |
| *bomVersion* | *int* | *-2147483648 to 2147483647* |
| *customerName* | *string* |  |
| *deliveryOrderNumber* | *string* |  |
| *id* | *long* | *-9223372036854775808 to 9223372036854775807* |
| *isAssembled* | *bool* |  |
| *isOnHold* | *bool* |  |
| *isOnShippingHold* | *bool* |  |
| *isPacked* | *bool* |  |
| *isScrapped* | *bool* |  |
| *materialName* | *string* |  |
| *materialPlantCode* | *string* |  |
| *plannedOrderNumber* | *string* |  |
| *productionOrderNumber* | *string* |  |
| *serialNumber* | *string* |  |
| *wipStatus* | *string* | *New / InProcess / Completed / Shipped / Archived* |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Retrieve WIP Attribute | | |
| **URL** | api/wips/{wipId}/attributes/ | | |
| **Http Method** | GET | | |
| **URL Segment** | **Name** | **Type** | **Remark** |
|  | *wipId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| **Parameter** | **Name** | **Type** | **Remark** |
| *attributeName* | *string* | *Optional* |
| **Return** | Single object with the following property | | |
| **Property Name** | **Type** | **Remark** |
| *wipAttributes* | *wipAttribute* | *Collection of objects* |
|  | | |
| wipAttribute | | |
| **Property Name** | **Type** | **Remark** |
| *id* | *long* | *-9223372036854775808 to 9223372036854775807* |
| *name* | *string* |  |
| *type* | *string* | *Boolean / Decimal / Integer / String* |
| *value* | *string* |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Insert WIP Attribute | | |
| **URL** | api/wips/{wipId}/attributes | | |
| **Http Method** | POST | | |
| **URL Segment** | **Name** | **Type** | **Remark** |
| *wipId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| **Parameter** | **Name** | **Type** | **Remark** |
| *wipAttributeName* | *string* | *Mandatory - max 50 characters* |
| *wipAttributeType* | *string* | *Mandatory - Boolean / Decimal / Integer / String* |
| *wipAttributeValue* | *string* | *Mandatory - max 100 characters* |
| **Return** | No object return | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Update WIP Attribute | | |
| **URL** | api/wips/{wipId}/attributes/{wipAttributeId} | | |
| **Http Method** | PUT | | |
| **URL Segment** | **Name** | **Type** | **Remark** |
| *wipId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| *wipAttributeId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| **Parameter** | **Name** | **Type** | **Remark** |
| *wipAttributeType* | *string* | *Mandatory - Boolean / Decimal / Integer / String* |
| *wipAttributeValue* | *string* | *Mandatory - max 100 characters* |
| **Return** | No object return | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Delete WIP Attribute | | |
| **URL** | api/wips/{wipId}/attributes/{wipAttributeId} | | |
| **Http Method** | DELETE | | |
| **URL Segment** | **Name** | **Type** | **Remark** |
| *wipId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| *wipAttributeId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| **Return** | No object return | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Start WIP | | |
| **URL** | api/wips/{wipId}/processSteps/start | | |
| **Http Method** | POST | | |
| **URL Segment** | **Name** | **Type** | **Remark** |
| *wipId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| **Parameter** | **Name** | **Type** | **Remark** |
| *resourceName* | *string* | *Mandatory* |
| **Return** | Single object with the following property | | |
| **Property Name** | **Type** | **Remark** |
| *wipProcessStepHistoryId* | *long* | *-9223372036854775808 to 9223372036854775807* |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Complete WIP Process Step | | |
| **URL** | api/wips/{wipId}/processSteps/{wipProcessStepHistoryId}/complete | | |
| **Http Method** | POST | | |
| **URL Segment** | **Name** | **Type** | **Remark** |
| *wipId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| *wipProcessStepHistoryId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| **Return** | Single object with the following properties | | |
| **Property Name** | **Type** | **Remark** |
| *inQueueRouteSteps* | *routeStep* | *Collection of objects* |
| *wipProcessStepHistoryStatus* | *string* | *Pass / Fail (has open failure)* |
|  | | |
| routeStep | | |
| **Property Name** | **Type** | **Remark** |
| *routeStepId* | *int* | *-2147483648 to 2147483647* |
| *routeStepName* | *string* |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | Abort WIP Process Step | | | |
| **URL** | api/wips/{wipId}/processSteps/{wipProcessStepHistoryId}/abort | | | |
| **Http Method** | POST | | | |
| **URL Segment** | **Name** | **Type** | **Remark** | |
| *wipId* | *long* | *-9223372036854775808 to 9223372036854775807* | |
| *wipProcessStepHistoryId* | *long* | *-9223372036854775808 to 9223372036854775807* | |
| **Return** | Single object with the following property | | | |
| **Property Name** | **Type** | | **Remark** |
| *inQueueRouteSteps* | *routeStep* | | *Collection of objects* |
|  | | | |
| routeStep | | | |
| **Property Name** | **Type** | | **Remark** |
| *routeStepId* | *int* | *-2147483648 to 2147483647* | |
| *routeStepName* | *string* |  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Insert Failure | | |
| **URL** | api/wips/{wipId}/processSteps/{wipProcessStepHistoryId}/symptoms | | |
| **Http Method** | POST | | |
| **URL Segment** | **Name** | **Type** | **Remark** |
| *wipId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| *wipProcessStepHistoryId* | long | -9223372036854775808 to 9223372036854775807 |
| **Parameter** | **Name** | **Type** | **Remark** |
| *failureMessage* | *string* | *Optional - if no value provided, will take symptom label as default value, unlimited characters* |
| *symptomLabel* | *string* | *Mandatory - max 100 characters* |
| **Return** | No object return | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Retrieve WIP Genealogy | | |
| **URL** | api/wips/{wipId}/genealogy | | |
| **Http Method** | GET | | |
| **URL Segment** | **Name** | **Type** | **Remark** |
| *wipId* | *long* | *-9223372036854775808 to 9223372036854775807* |
| **Parameter** | **Name** | **Type** | **Remark** |
| *maxLevel* | *int* | *1 to 10 - if other value is given or no value provided, will take 10 as default value* |
| *serializedOnly* | *string* | *True / False - if other value is given or no value provided, will take False as default value* |
| **Return** | Single object with the following property | | |
| **Property Name** | **Type** | **Remark** |
| *wipAssembleHistoryList* | *wipAssembleHistory* | *Collection of objects* |
|  | | |
| wipAssembleHistory | | |
| **Property Name** | **Type** | **Remark** |
| *assembledDate* | *string* |  |
| *assembledRouteStep* | *string* |  |
| *crd* | *string* |  |
| *id* | *int* | *-2147483648 to 2147483647* |
| *manufacturerPartNumber* | *string* |  |
| *materialId* | *int* | *-2147483648 to 2147483647* |
| *materialName* | *string* |  |
| *materialPlantCode* | *string* |  |
| *operatorUserLogin* | *string* |  |
| *parentId* | *int* | *-2147483648 to 2147483647* |
| *phoenixMaterialType* | *string* |  |
| *quantity* | *decimal* |  |
| *returnCount* | *int* | *-2147483648 to 2147483647* |
| *serializedMaterialId* | *long* | *-9223372036854775808 to 9223372036854775807 - Nullable* |
| *serialNumber* | *string* |  |
| *wipId* | *long* | *-9223372036854775808 to 9223372036854775807 - Nullable* |
| *dataCollectionList* | *dataCollection* | *Collection of objects* |
| *uniqueProductDataList* | *uniqueProductData* | *Collection of objects* |
|  | | |
| dataCollection | | |
| **Property Name** | **Type** | **Remark** |
| *dataType* | *string* |  |
| *labelName* | *string* |  |
| *lowerLimit* | *string* |  |
| *status* | *string* |  |
| *unitOfMeasurement* | *string* |  |
| *upperLimit* | *string* |  |
| *value* | *string* |  |
|  | | |
| uniqueProductData | | |
| **Name** | **Type** | **Remark** |
| *name* | *string* |  |
| *type* | *string* |  |
| *value* | *string* |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | Insert serialized material | | |
| **URL** | api/SerializedMaterials | | |
| **Http Method** | POST | | |
| **Parameter** | Collection of objects with the following property | | |
| **Name** | **Type** | **Remark** |
| *customerName* | *string* | *Optional* |
| *materialName* | *string* |  |
| *plantCode* | *string* |  |
| *serialNumber* | *string* |  |
| **Return** | Single object with the following property | | |
| **Property Name** | **Type** | **Remark** |
| *serializedMaterials* | *serializedMaterial* | *Collection of objects* |
|  | | |
| *serializedMaterial* | | |
| **Property Name** | **Type** | **Remark** |
| *customerName* | *string* |  |
| *materialName* | *string* |  |
| *plantCode* | *string* |  |
| *serialNumber* | *string* |  |
| *errorMessages* | *String array* | *Failed if it’s not empty, and it contain the fail reason*  *Success if it’s empty.* |

NOTE:

To call the API, a security token is needed. In order to retrieve a security token, we can call a API at api/tokens, and passing an active user account and password. Then create a Http header entry with "**Authorization**" name, and assign it with value "**Bearer** " + security token. The active user account must be associated to "**Can Access From External Client Gateway**" permission, in order to allow access to API.

Any validation or exception thrown, will return a collection of objects with the following structure.

Error

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Remark** |
| message | *String array* |  |

C# SAMPLE CODE:

Encrypt password to retrieve security token

using System;

using System.Security.Cryptography;

using System.Text;

public class Cipher

{

public string Encrypt(string value)

{

var aes = new AesManaged()

{

KeySize = 128,

BlockSize = 128,

Mode = CipherMode.CBC,

Padding = PaddingMode.PKCS7,

Key = Convert.FromBase64String("YGjDImfP5bf8F7kGbdqWcA==*")*,

IV = Convert.FromBase64String("bDA11yRcbUsTC9jl5LnuiQ==")

};

byte[] bytes = Encoding.UTF8.GetBytes(value);

ICryptoTransform encryptor = aes.CreateEncryptor();

byte[] ctext = encryptor.TransformFinalBlock(bytes, 0, bytes.Length);

string text = Convert.ToBase64String(ctext);

return text;

}

}

Invoke Get WIP Attributes function (using RestSharp nuget package)

using System.Collections.Generic;

public class WipAttributeList

{

public List<WipAttribute> WipAttributes { get; set; }

public class WipAttribute

{

public long Id { get; set; }

public string Name { get; set; }

public string Type { get; set; }

public string Value { get; set; }

}

}

using RestSharp;

public class User

{

public string GetToken(string userLogin, string password)

{

var cipher = new Cipher();

string encryptedPassword = cipher.Encrypt(password);

var request = new RestRequest();

request.Method = Method.GET;

request.Resource = "api/tokens";

request.AddQueryParameter("userLogin", userLogin);

request.AddQueryParameter("password", encryptedPassword);

var client = new RestClient("http://stacktest");

UserToken userToken = client.Execute<UserToken>(request).Data;

return userToken.Token;

}

public class UserToken

{

public string Token { get; set; }

}

}

using RestSharp;

public class WipAttributeRepository

{

public WipAttributeList Get(long wipId)

{

var user = new User();

string token = user.GetToken("userLogin", "password");

string header = string.Format("Bearer {0}", token);

var request = new RestRequest();

request.Method = Method.GET;

request.Resource = "api/wips/{wipId}/attributes";

request.AddUrlSegment("wipId", wipId.ToString());

request.AddHeader("Authorization", header);

var client = new RestClient("http://stacktest");

WipAttributeList wipAttributes = client.Execute<WipAttributeList>(request).Data;

return wipAttributes;

}

}

JAVA SAMPLE CODE TO ENCRYPT PASSWORD

import java.util.Base64;

import javax.crypto.Cipher;

import javax.crypto.spec.IvParameterSpec;

import javax.crypto.spec.SecretKeySpec;

class encrypt {

public static void main(String[] args) {

try {

String input = "password";

IvParameterSpec iv = new IvParameterSpec(Base64.getDecoder().decode("bDA11yRcbUsTC9jl5LnuiQ=="));

SecretKeySpec key = new SecretKeySpec(Base64.getDecoder().decode("YGjDImfP5bf8F7kGbdqWcA=="), "AES");

Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");

cipher.init(Cipher.ENCRYPT\_MODE, key, iv);

byte[] bytes = cipher.doFinal(input.getBytes("UTF-8"));

String password = Base64.getEncoder().encodeToString(bytes);

System.out.println("Encrypted password: " + password);

}

catch (Exception ex) {

ex.printStackTrace();

}

}

}