Smarten
Augmented Analytics
Powered by ElegantJ BI

Integration with your App

Version 5.1

Document Version: 8.0
Product Version: 5.1

Document Information	
Document ID	Smarten-Integration-With-Your-App
Document Version	8.0
Product Version	5.1
Date	02-December-2020
Recipient	NA
Author	EMTPL

Statement of Confidentiality, Disclaimer and Copyright

This document contains information that is proprietary and confidential to EMTPL, which shall not be disclosed, transmitted, or duplicated, used in whole or in part for any purpose other than its intended purpose. Any use or disclosure in whole or in part of this information without the express written permission of EMTPL is prohibited.

Any other company and product names mentioned are used for identification purpose only, may be trademarks of their respective owners and are duly acknowledged.

Disclaimer

This document is intended to support administrators, technology managers or developers using and implementing Smarten. The business needs of each organization will vary and this document is expected to provide guidelines and not rules for making any decisions related to Smarten. The overall performance of Smarten depends on many factors, including but not limited to hardware configuration and network throughput.

Contents

1 Introduction 4

2 Embed BI objects or Create new object wizard using API 4

2.1 Integration Flow 4
2.2 Integration Methods 5
2.2.1 Use JavaScript API 5
2.2.1.1 Embed BI Object 5
2.2.1.2 Embed "Create New Object" Wizard 7
2.2.2 Use RESTful Web Services API 9
2.3 Encryption 14

3 Single Sign-On Integration 15

3.1 Integration Flow 15
3.2 Configuration 16

4 Administration and Configuration API 17

4.1 Introduction 17
4.2 Integrating Web Services 17
4.3 User and Role Management Service 18
4.4 Repository Management Service 25
4.5 Cube Management Service 31
4.6 Tenant Management Service 40

5 SmartenInsight model integration API 43

5.1 Get model data 43
5.2 Get prediction for single record (Apply) 44
5.3 Get prediction for multiple records (Mass apply) 47
5.4 Get model interpretation 49
5.5 Get model summary information 50

6 Product and Support Information 50

1 Introduction

Smarten provides easy-to-implement options to seamlessly integrate Smarten or its various BI objects (Dashboards, Crosstab, Tabular, KPIs and Graphs) into your application. There are two options available to integrate Smarten with your application.

- Embed BI objects / create new object wizard using API
- SSO integration

This document also covers integration for administration and configuration functions.

2 Embed BI objects or Create new object wizard using API

You can embed views of Smarten objects into your web pages, web applications, and extranet or intranet portals. The embedded views will integrate seamlessly, show updated data in the objects, and provide complete interaction to users. Smarten objects can be embedded into your browser-based applications that use Java, .NET, ASP, PHP, and most of the other web technologies.

You can also request Smarten object data using this API into any web, desktop, or mobile technologies supporting RESTful web services.

Integration through API will follow the same licensing and access rights used on the Smarten server.

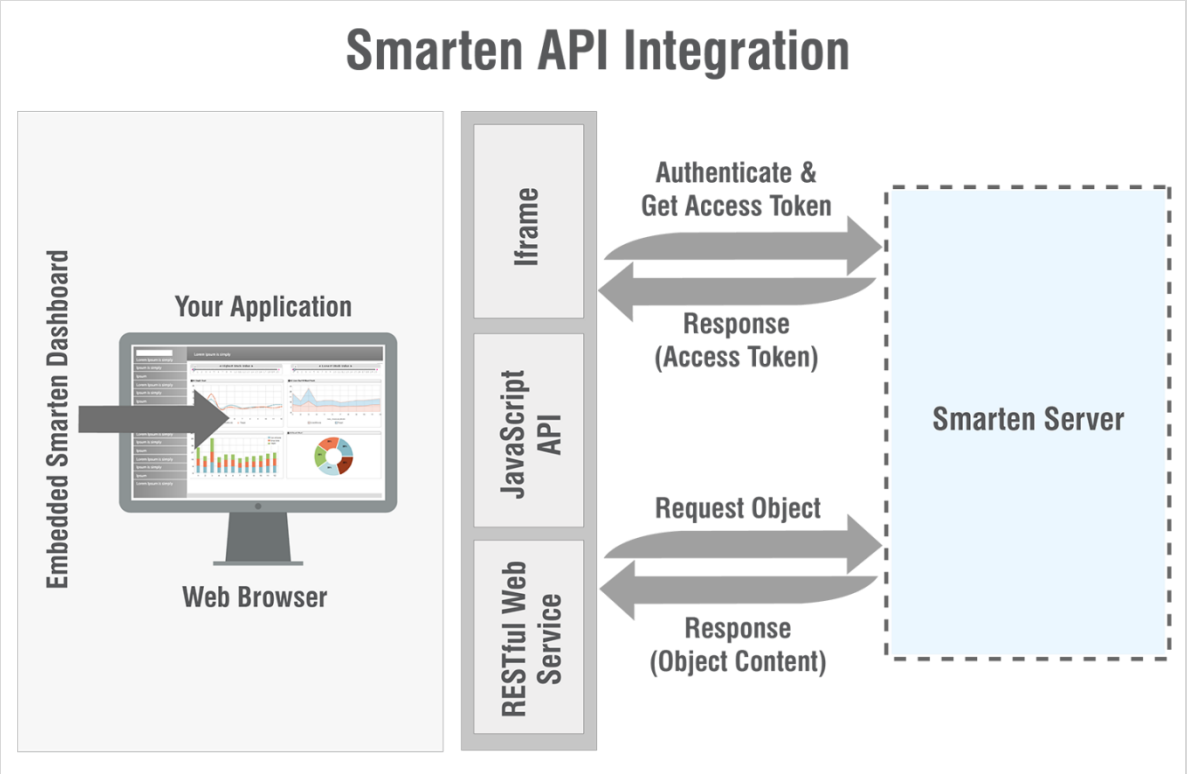
Smarten provides multiple technical approaches for integration, and with minimal technical skills and effort, you can start embedding Smarten objects in your applications.

2.1 Integration Flow

Smarten provides industry-standard integration flow to seamlessly and securely integrate Smarten objects with your applications.

All Smarten users and BI objects are assigned system-generated unique keys that are accessible from the Administration interface. Unique keys for Smarten users are called Access Keys, and unique keys for Smarten objects are called Object IDs.

Higher-level Integration Flow is as described below:



EMBED BI OBJECTS - HIGHER-LEVEL INTEGRATION FLOW

Step1: Request for Object is sent to the Smarten Server with Object ID and Access Key or login credentials.

Step2: The Smarten server will authenticate credentials and return a unique, encrypted Access Token to the requestor. The Access Token expires after expiry interval, and request is verified against defined scope. The Administrator can define scope parameters to restrict requests from a particular IP address, a range of IP addresses, a domain name, or a list of domain names.

Step3: Request for Object is sent to Smarten Server, with Access Token

Step4: The Smarten server will verify Access Token and return Smarten object contents.

2.2 Integration Methods

You can embed Smarten objects in two ways:

2.2.1 Use JavaScript API

2.2.1.1 Embed BI Object

You can use Smarten JavaScript objects in your web application code with required parameters, such as URL of Smarten Server, Access Key for the user, Object ID for Smarten object, and Toolbar preference.

JavaScript API Parameters:

Parameter Name	Mandatory	Description
url	Yes	Smarten server URL

objectid	Yes	Object ID of the BI object you are requesting. You will be able to find the Smarten Object ID for each object from the Administration interface.
accesskey	No	You can use the Access Key or username/password. You will be able to find the Access Key for each user from the Administration interface.
username	No	Smarten login user name
password	No	Smarten login password
ssoParams	No	This parameter is used when Smarten is configured with SSO authentication. You can pass SSO authentication parameters through this attribute. For example, *"srvUser=admin&srvSessionId=adws4a7d7sfeka452&param1=value1&param2=value2"*
toolbar	No	Parameter to show/hide object toolbar Possible values: true/false Default value: false
showobjectname	No	Parameter to show/hide object name Possible values: true/false Default value: false
containerid	No	Parameter to specify HTML container element (such as) where you want to embed and render BI object.
tokenid	No	Parameter for the Access Token generated by the Smarten server. This parameter will be used only when you have generated the Access Token from the Smarten server through Web Services API and you want to embed an object using JavaScript in your web-based application. Please read notes in Web Services API for more details.
width	No	Object width in your HTML container
height	No	Object height in your HTML container
adaptive	No	Parameter to enable/disable object adaptive behaviour Possible values: true/false Default value: false
teamup	No	Parameter to show/hide TeamUp icon in object Possible values: true/false Default value: false
<column_name>	No	User can pass column name as parameter and filter values as parameter string to filter data for object For example, if user want to filter data based on State column, parameter will be passed as shown below: State: "Gujarat$*$Maharastra" Where $*$ is separator between values to be filtered.

Code Example:

```
<script type="text/javascript" src="http://<SMARTENURL>/js/EJBIapi.js"></script>
<div style="height:500px; width:500px;" id="test">
    <script type="text/javascript">
    varparams = {
        url:"<SMARTENURL>", // required.
        objectid:"<ObjectId>",
        accesskey:"<accessKey>",  //  alternate to username – password and ssoParams
        username:"<username>", password:"<password>", // alternate to accesskey and ssoParams
        ssoParams:"<SSO Configuration Parameters>", // alternate to accesskey and username-password
        toolbar:"true",
        showobjectname:"true",
        containerid:"<container element Id>",
        tokenid:"<TokenKey>", // alternate all expect url.
        width:"<width of iframe>",
        height:"<height of iframe>",
        adaptive:"true",
        teamup:"true",
            State:"Gujarat$*$Maharastra",
            City:"Ahmedabad$*$Pune"
    };
        newEJBIapi(params).loadObject();
    </script>
</div>
```

2.2.1.2 Embed "Create New Object" Wizard

You can use JavaScript API to create new Smarten objects from your web application with required parameters, such as URL of Smarten Server, Access Key for the user and Smarten object Type.

JavaScript API Parameters:

Parameter Name	Mandatory	Description
url	Yes	Smarten server URL
accesskey	No	You can use the Access Key or username/password. You will be able to find the Access Key for each user from the Administration interface.

username	No	Smarten login user name
password	No	Smarten login password
ssoParams	No	This parameter is used when Smarten is configured with SSO authentication. You can pass SSO authentication parameters through this attribute. For example, *"srvUser=admin&srvSessionId=adws4a7d7sfeka452&param1=value1&param2=value2"*
objectType	Yes	CROSSTAB=0,GRAPH=1,Tabular=2,Dashboard=3,KPI=4,KPIGroup=5, GeoMap=6, SmartenView=7 You can use Constants from EJBIAPI js As shown in Example.
containerid	No	Parameter to specify HTML container element (such as) where you want to embed and render BI object.
tokenid	No	Parameter for the Access Token generated by the Smarten server. This parameter will be used only when you have generated the Access Token from the Smarten server through Web Services API and you want to embed an object using JavaScript in your web-based application. Please read notes in Web Services API for more details.
width	No	Object width in your HTML container
height	No	Object height in your HTML container

Code Example:

```
<script type="text/javascript" src="http://<SMARTENURL>/js/EJBIapi.js"></script>
<div style="height:500px; width:500px;" id="test">
        <script type="text/javascript">
                varparams = {
                        url:"<SMARTENURL>", // required.
                        objectType:"<Objecttype>",   // required
                        accesskey:"<accessKey>",  //  alternate to username – password and ssoParams

                        username:"<username>", password:"<password>", // alternate to accesskey and ssoParams
                        ssoParams:"<SSO Configuration Parameters>", // alternate to accesskey and username-password
                        containerid:"<container element Id>",
                        tokenid:"<TokenKey>" // alternate all expect url.
                        width:"<width of iframe>",
                        height:"<height of iframe>",
                };
varEJBIAPI_Obj = new EJBIapi(params);
```

```
        varobjTypeId = -1;
        switch(val){
                case 'Analysis' :
                        objTypeId = EJBIAPI_Obj.CROSSTAB;
                        break;
                case 'Graph' :
                        objTypeId = EJBIAPI_Obj.GRAPH;
                        break;
                case 'Dashboard' :
                        objTypeId = EJBIAPI_Obj.DASHBOAR;
                        break;
                case 'Tabular' :
                        objTypeId = EJBIAPI_Obj.TABULAR;
                        break;
        }
                EJBIAPI_Obj.newObject(objTypeId);
        </script>
</div>
```

2.2.2 Use RESTful Web Services API

You can call the Smarten Web Services URL from your application code with required parameters, such as the Access Key for the user, the Object ID for the Smarten object, and Toolbar preference.

Web Services API URLs:

1. **Get Access Token**

   http://<EJBI_URL>/API/getToken

Web Services API Parameters:

Parameter Name	Mandatory	Description
objectid	Yes	Object ID of the BI object you are requesting. You will be able to find the Smarten Object ID for each object from the Administration interface
accesskey	No	You can use the Access Key or username/ password. You will be able to find the Access Key for each user from the Administration interface.
username	No	Smarten login user name
password	No	Smarten login password
toolbar	No	Parameter to show/hide object toolbar Possible values: true/false Default value: false
showobjectname	No	Parameter to show/hide object name Possible values: true/false Default value: false
type	No	Parameter to get an object's UI or XML data. The XML data option is not applicable for Dashboard objects in

		Smarten. Possible Values: object – for object UI, data – for XML data Default Value: object
<column_name>	No	User can pass column name as parameter and filter values as parameter string to filter data for object For example, if user want to filter data based on State column, parameter will be passed as shown below: State=Gujarat$*$Maharastra Where $*$ is separator between values to be filtered.

> Note: Once you obtain the Access Token with this API call, you have two options:
> Option 1—Use "Get Object content" web services API to get object content from the Smarten server
> Option 2—Use "JavaScript API" with "tokenID" parameter to embed object in your web application

2. **Get Object content**

http://<EJBI_URL>/API/getObject

Web Services API Parameters:

Parameter Name	Mandatory	Description
tokenid	Yes	Token generated and returned by Smarten server as a result of "Get Access Token" request
adaptive	No	Parameter to enable/disable object adaptive behaviour Possible values: true/false Default value: false
teamup	No	Parameter to show/hide TeamUp icon in object Possible values: true/false Default value: false

Code Examples:

JAVA

```
try{
        String ejbiServerUrl = "SMARTENURL";
        String getTokenUrl = "/API/getToken";
        String getResourceUrl = "/API/getObject ";

        // Getting Token
        URL obj = newURL(ejbiServerUrl+getTokenUrl);
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();
        con.setRequestMethod("POST");
        String urlParameters =
"username=<username>&password=<password>&objectid=<objectId>&toolbar=<true/false>&type=
<obejct/data>&url<SMARTENURL-
```

```
required>&containerid=<containerId>&tokenid=<TokenKey>&height=<iframeHeight>&width=<ifram
eWidth>&state=Gujarat$*$Maharashtra";

        // Set Request Header

        /**  Addingclien info in Request**/
        String originIp = request.getHeader("X-FORWARDED-FOR");
        if (originIp == null) {
                originIp = request.getRemoteAddr();
        }
        String userAgent = request.getHeader("User-Agent");

        con.setRequestProperty("X-FORWARDED-FOR", originIp);
        con.setRequestProperty("user-agent", userAgent);
        /** END Adding clien info in Request**/

        // Send post request
        con.setDoOutput(true);
        DataOutputStreamwr = newDataOutputStream(con.getOutputStream());
        wr.writeBytes(urlParameters);
        wr.flush();
        wr.close();

        BufferedReader in = newBufferedReader(
newInputStreamReader(con.getInputStream()));
        String inputLine;
        StringBuffer token = newStringBuffer();
        while ((inputLine = in.readLine()) != null) {
                token.append(inputLine);
        }
        in.close();
        //  Getting Object Content

        String url = ejbiServerUrl+getResourceUrl+"?       tokenid="+token.toString().trim();
        String output= "<iframe src='"+url+"' height='100%' width='100%' ></iframe>";
}
catch (Exception e) {
        e.printStackTrace();
}
```

PHP

```
<?php
//Variables for API URLs
$strEjbiServerUrl = "<SMARTENURL>";
$strGetTokenUrl = "/API/getToken";
$strGetResourceUrl = "/API/getObject ";
$strKey="<saltKey>";
```

```
// Get Access Token
$strUrlParameters =
"username=".EncryptText("<username>",$strKey)."&password=".EncryptText("<password>",$strKey)
."&objectId=<objectId>&toolbar=<true/false>";
///Generate Web request for Authentication Token
$ch = curl_init();
 /**  Adding clien info in Request**/

if(isset($_SERVER['HTTP_X_FORWARDED_FOR']) && trim($_SERVER['HTTP_X_FORWARDED_FOR'])
!= ""){
   $ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
}else{
   $ip = $_SERVER['REMOTE_ADDR'];
}

$ua = $_SERVER['HTTP_USER_AGENT'];

curl_setopt($ch, CURLOPT_HTTPHEADER, array("REMOTE_ADDR: $ip","X_FORWARDED_FOR: $ip"));
curl_setopt($ch,CURLOPT_USERAGENT,$ua);

/** END Adding clien info in Request**/

curl_setopt($ch, CURLOPT_URL, $strEjbiServerUrl.$strGetTokenUrl);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $strUrlParameters);
$strToken = curl_exec($ch);
if(curl_errno($ch)){
echo 'CurlToken Error: ' . curl_error($ch);
}
curl_close($ch);
//Retrive Authentication Token
$strUrlParameters = "?tokenid=".trim($strToken);
?>
```

.NET

```
public string strToken;
try
      {
         // Variables for API URLs
stringstrEjbiServerUrl = "SMARTENURL";
stringstrGetTokenUrl = "/API/getToken";
stringstrGetResourceUrl = "/API/getObject ";
stringstrKey = "<saltKey>";
         // Hashtable for Request Parameters

         Dictionary<string,string>dicRequestParameters= new Dictionary<string,string>();
         // Post Parameters with Encrypted Parameters
```

```
dicRequestParameters.Add("username", EncryptText("<username>", strKey));
dicRequestParameters.Add("password", EncryptText("<password>", strKey));
dicRequestParameters.Add("objectId", "<objectId>");
stringstrPostData = string.Empty;
        // Apply Url Post
foreach (string key in dicRequestParameters.Keys)
        {
strPostData += HttpUtility.UrlEncode(key) + "=" + HttpUtility.UrlEncode(dicRequestParameters[key])
+ "&";
        }
        //  Request for Get Access Token
HttpWebRequestobjWebRequest = (HttpWebRequest)WebRequest.Create(strEjbiServerUrl +
strGetTokenUrl);
       // set Request Header
        /**  Addingclien info in Request**/
        String originIp = Request.ServerVariables["HTTP_X_FORWARDED_FOR"];
if (originIp == null)
        {
originIp = Request.ServerVariables["REMOTE_ADDR"];
        }
        String userAgent = Request.UserAgent;
objWebRequest.UserAgent = userAgent;
objWebRequest.Headers.Add("X-FORWARDED-FOR", originIp);
        /** END Adding clien info in Request**/
objWebRequest.Method = "Post";
objWebRequest.ContentType = "application/x-www-form-urlencoded";
byte[] byteArray = Encoding.ASCII.GetBytes(strPostData);
objWebRequest.ContentLength = byteArray.Length;
        Stream dataStream = objWebRequest.GetRequestStream();
dataStream.Write(byteArray, 0, byteArray.Length);
dataStream.Close();
        // Response for Authentication Token
HttpWebResponseobjWebResponse = (HttpWebResponse)objWebRequest.GetResponse();
dataStream = objWebResponse.GetResponseStream();
StreamReader reader = new StreamReader(dataStream);
strToken = reader.ReadToEnd();
objWebResponse.Close();
        // Check First Request response
if (!string.IsNullOrEmpty(strToken))
        {
HttpContext.Current.Response.Write("<iframe src='" + strEjbiServerUrl + strGetResourceUrl +
"?tokenid=" + strToken.ToString().Trim() + "' height='100%' width='100%' ></iframe>");
        }
      }
catch (Exception ex)
      {
HttpContext.Current.Response.Write(ex.Message);
      }
```

2.3 Encryption

Smarten API provides an encryption mechanism for secure transmission of important parameters, such as the Access Key and login credentials. The Administrator can set a preference for encryption (yes or no) from the administration panel. If the preference is set for encryption, you must use the AES (AES/ECB/NoPadding) encryption algorithm to encrypt these parameters (Access Key and login credentials) in your API calls. You can obtain secure encryption salt (secret key) from the administration panel.

Code Examples:

JAVA

```
public static String encrypt(String strToEncrypt, byte[] key) {
try {
        Cipher cipher = Cipher.getInstance("AES/ECB/NoPadding");
cipher.init(Cipher.ENCRYPT_MODE, new SecretKeySpec(key, "AES"));
byte[] text = Arrays.copyOf(strToEncrypt.getBytes("UTF-8"),
((strToEncrypt.getBytes("UTF-8").length/16)+1)*16);
return (Base64.encodeBase64String(cipher.doFinal(text)));
    } catch (Exception e) {
System.out.println("Error while encrypting: " + e.toString());
    }
return null;
  }
```

PHP

```
functionEncryptText($strToEncrypt,$strKey){
        $cipher=MCRYPT_RIJNDAEL_128;
        $mode=MCRYPT_MODE_ECB;
$returnVal=trim(base64_encode(mcrypt_encrypt($cipher, $strKey, $strToEncrypt, $mode,
mcrypt_create_iv(mcrypt_get_iv_size($cipher, $mode), MCRYPT_RAND))));
        return $returnVal;
 }
```

.NET

```
public String EncryptText(String plainText, String key)

  {

varplainBytes = Encoding.UTF8.GetBytes(plainText);

varkeyBytes = new byte[16];

varsecretKeyBytes = Encoding.UTF8.GetBytes(key);

Array.Copy(secretKeyBytes, keyBytes, Math.Min(keyBytes.Length, secretKeyBytes.Length));

varobjRijndaelManaged= new RijndaelManaged

    {

      Mode = CipherMode.ECB,
```

```
        Padding = PaddingMode.Zeros,
KeySize = 128,
BlockSize = 128,
        Key = keyBytes,
        IV = keyBytes
    };
return Convert.ToBase64String(objRijndaelManaged.CreateEncryptor()
        .TransformFinalBlock(plainBytes, 0, plainBytes.Length));
  }
```

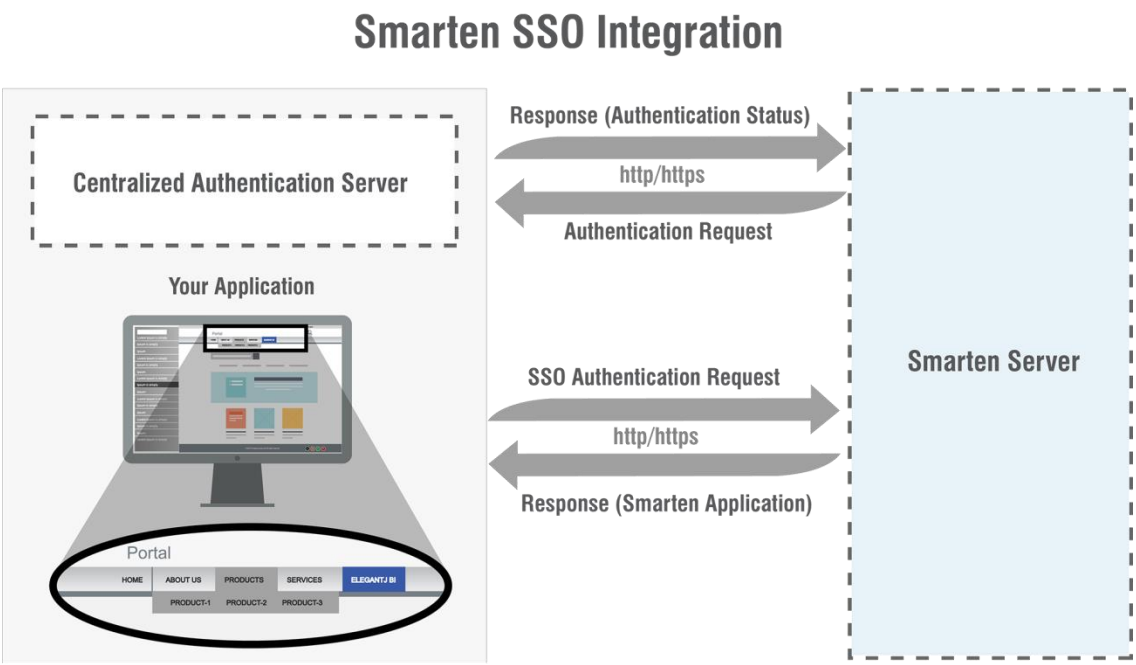
3 Single Sign-On Integration

Smarten provides API for Single Sign-On (SSO) authentication integration with other applications. Developers can easily implement their custom SSO authentication mechanism to link Smarten with their existing portal or application.

3.1 Integration Flow

Smarten provides industry-standard integration flow using SSO (oAuth) to seamlessly and securely integrate Smarten with your applications.

Higher-level integration flow is as described below:



SSO INTEGRATION - HIGHER-LEVEL INTEGRATION FLOW

Step1: When user clicks on “Smarten” link from your application, SSO authentication request will be sent to Smarten server using HTTP GET/POST request. User authentication credentials such as username and access token are passed with this request. Generally these parameter values are encrypted.

Smarten SSO authentication request URL is as follows:

http://<Smarten server URL>/SSO/login
(e.g. http://10.0.0.10:8080/Smarten/SSO/login)

Step2: Smarten receives parameters and sends these parameters to SSO authentication server to authenticate user.

Step3: SSO authentication server authenticates credentials and sends authentication status to Smarten server as response.

Step4: Based on SUCCESS or FAILED status, Smarten will provide access and authenticated user will get access of Smarten as per access rights configured in Smarten.

3.2 Configuration

SSO integration related settings are provided in ssoConfig.properties file in "conf" folder of Smarten deployment.

SSO integration API configuration parameters:

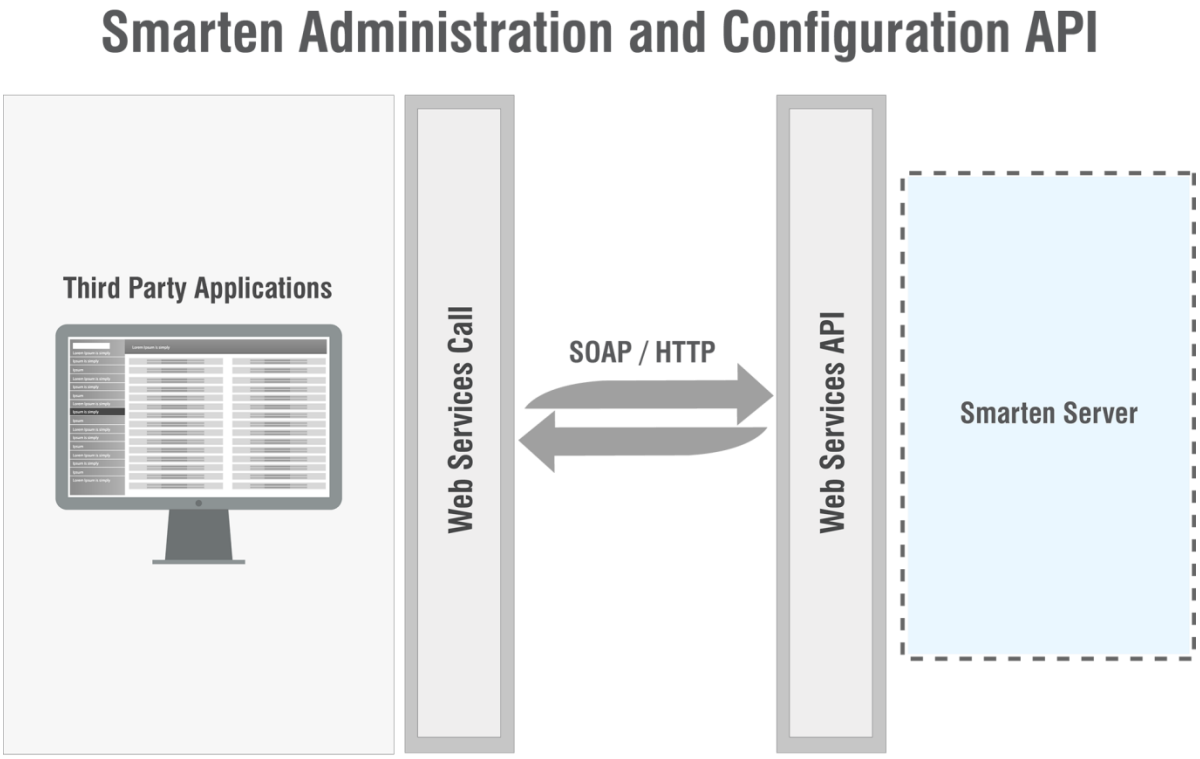
Parameter Name	Description
sso.enable	This parameter is used to enable or disable SSO feature. Possible values: true / false
sso.authServerURL	Parameter to specify SSO authentication URL
sso.result.sucess	Parameter to specify SSO authentication success result value
sso.result.failed	Parameter to specify SSO authentication failure result value
sso.params.username	Parameter to specify user ID parameter name in SSO authentication request from your portal.
sso.encryption	Parameter to indicate whether the encryption is ON or OFF.
sso.encryption.saltkey	If encryption is enabled, this salt key is used for decryption based on encryption algorithm.
sso.encryption.algo.class	Parameter to specify encryption/decryption algorithm class. Smarten provides default encryption class based on Base64 encryption algorithm. However, you can implement your own custom encryption logic. You can implement com.Smarten.api.encrypt.EncryptionProvider interface and put it in deployment classpath. After that you need to specify your class name in this parameter.
sso.authentication.provider.class	Parameter to specify SSO authentication provider class. Smarten provides default implementation as per integration flow mentioned above. However, depending up on your authentication server implementation, you can implement your custom logic. You need to implement com.Smarten.security.AuthenticationProvider interface as per requirement.

4 Administration and Configuration API

4.1 Introduction

Administration and Configuration Web services API is used to manage Smarten configuration and administration related tasks from third-party applications.

Web services are SOAP based secure calls and can be called from most of the popular programming languages such as .NET, Java and PHP.



CONFIGURATION AND ADMINISTRATION API - HIGHER-LEVEL INTEGRATION FLOW

4.2 Integrating Web Services

To integrate Smarten configuration and administration web services, developers need to follow standard SOAP web service integration process. Developers need to generate functional stubs through WSDL definitions.

The functional stubs will allow developers to make standard function calls in their native programming language which will directly communicate with the web services provided by Smarten. The process of creating functional stubs will also generate objects required by the web service.

Smarten web services are secured and requires authentication for integration. It uses HTTP basic authentication method for accessing the web service. Developers need to provide credentials of Smarten user for authentication.

Smarten web services WSDL definitions can be found from below URLs:

- User and Role Management service
  http://<SMARTENURL>/ws/userRoleManagement.wsdl
- Repository Management service
  http://<SMARTENURL>/ws/repositoryManagement.wsdl

- Cube Management Service
  http://<SMARTENURL>/ws/cubeManagement.wsdl
- Tenant Management Service
  http://<SMARTENURL>/ws/tenantManagement.wsdl

4.3 User and Role Management Service

This service will be used to replicate or synchronize users and roles from third party application to Smarten. This allows Smarten to identify the user who is logged in, and also to apply required access permissions to specific users and roles. Synchronization is usually performed using web service calls from the third party application to Smarten.

Web service functions:

Functions	Description
addRole	This function will create roles
updateRole	This function will modify role details
deleteRole	This function will delete rows
addUser	This function will create users
updateUser	This function will modify users
deleteUser	This function will delete users
getRoles	Retrieve list of roles
getUsers	Retrieve list of users
getUserAccessKey	Retrieve list of user access keys
chnagePassword	This function will change password of user

addRole

This function will create roles

Request Element	**Data Type**	**Description**	**Setting Code**
roleInfo	List of RoleInfo	The RoleInfo object holding all of the new role's details	requestObject.getRoleInfo.add(RoleInfo)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

updateRole

This function will update roles

Request Element	Data Type	Description	Setting Code
roleInfo	List of RoleInfo	The RoleInfo object holding all of the role's details	requestObject.getRoleInfo.add(RoleInfo)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

deleteRole

This function will delete roles

Request Element	Data Type	Description	Setting Code
roleName	String	Rolename of the role to be deleted	setRoleName()

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

addUser

This function will create users

Request Element	Data Type	Description	Setting Code
userInfo	List of UserInfo	The UserInfo object holding all of the new user's details	getUserInfo.add(UserInfo)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

updateUser

This function will update users

Request Element	Data Type	Description	Setting Code
userInfo	List of UserInfo	The UserInfo object holding all of the user's details	getUserInfo.add(UserInfo)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

deleteUser

This function will delete users

Request Element	Data Type	Description	Setting Code
userName	List of String	Username of the role to be deleted	getUserName().add(userName)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

getRoles

Retrieve list of roles OR Details of specific Role when role name is given

Request Element	Data Type	Description	Setting Code
roleName	String	Name of the role**(optional)** which information is desired, if role name is not given then list of roles will be returned	setRoleName()

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()
roles	List of RoleInfo	List of roles for the web service call	getRoles()

getUsers

Retrieve list of users **OR** Details of specific User when user name is given

Request Element	Data Type	Description	Setting Code
userName	String	Name of the user**(Optional)** which information is desired, if user name is not given then list of users will be returned	setUserName()

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call	getStatus()
users	List of UserInfo	List of users for the web service call	getUsers()

getUserAccessKey

Retrieve list of user access keys **OR** Specific user access key when user name is given

Request Element	Data Type	Description	Setting Code
userName	String	Name of the user**(Optional)** which information is desired, if user name is not given then list of user access keys will be returned	setUserName()

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()
keys	List of ApiUserClientInfo	List of users for the web service call	getKeys()

changePassword

This function will change password of user

Request Element	Data Type	Description	Setting Code
userDetailList	List of ChangePasswordUserDetails	Name of the users which passwords required to be changed.	getUserDetailList().add(changePasswordUserDetailsObject)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()
users	UserInfo	List of users for the web service call	getUsers()

User and Role Management Web Service Elements:

RoleInfo Element

These are the parameters that you need to set in the RoleInfo object:

Attribute Name	Data Type	Description	Setting Code
roleName	String	Role name for new role	setRoleName()
roleDesc	String	Role description for new role	setRoleDescription()
roleFunctionalRights	List of RoleFunctionalRightsInfo	Functional rights for new role	getRoleFunctionalRights().add(newItem)

RoleFunctionalRightsInfo

These are the parameters that you need to set in the RoleFunctionalRightsInfo object:

Attribute Name	Data Type	Description	Setting Code
moduleId	Integer	Folder = 1 Tabular = 2 Query = 3 Scheduler = 4	SetModuleId()

		Database Profile = 5 Data Model = 6 Cube = 7 Analysis = 8 Graph = 9 Export = 10 Analysis Views = 11 Dashboard = 12 Scorecard = 13 Dashboard Template= 14 Impact Analysis =15 KPI = 16 KPI Group = 17	
permissions	Integer	Permissions to be assign for the role. 1 = view 2 = create 3 = view & create 4 = delete 5 = view & delete 6 = create & delete 7 = view, create & delete	setPermission()

UserInfo

These are the parameters that you need to set in the UserInfo object:

Attribute Name	Data Type	Description	Setting Code
userTypeBI	Integer	Type of the user. Possible Values : 1. Administrator 2. Power User 3. User 4. Publish Only User	setUserTypeBI()
userActive	Boolean	Specify whether the user will be active or not (i.e. true for active)	setUserActive()
password	String	Password for new user	setPassword()
emailId	String	email address for new user	setEmailId()

roleNames	List of RoleInfo	Name of roles to be assigned to the user	getRoleInfo.add(RoleInfo)
username	String	username for new user	setUsername()
personName	String	Person name for new user	setPersonName()
deptName	String	Department for new user	setDeptName()

ApiUserClientInfo

These are the parameters that you need to set in the ApiUserClientInfo object:

Attribute Name	Data Type	Description	Setting Code
secretKey	String	secret access key for the user	setSecretKey()
userName	String	user name of the user	setUserName()
userId	String	user id of the user	setUserId()

ChangePasswordUserDetails

These are the parameters that you need to set in the ChangePasswordUserDetails object:

Attribute Name	Data Type	Description	Setting Code
username	String	username for which password needs to be changed	setUsername()
password	String	new password to be set	setPassword()

4.4 Repository Management Service

This service will be used to manage access permissions of various entities such as BI objects, cubes, folders etc.

Web service functions:

Functions	Description
setFolderPermission	This function will be used to set folder permissions

Functions	Description
setObjectPermission	This function will be used to set object permissions
getFolderList	Retrieves list of folders in repository
getObjectList	Retrieves list of objects in repository
addFolder	This function will be used to add new folder
deleteFolder	This function will be used to delete folder
deleteObject	This function will be used to delete object

Repository Management Web Service Functions

setFolderPermission

This function will used to set folder permissions

Request Element	**Data Type**	**Description**	**Setting Code**
folderPermission	List of FolderPermission	Folder permission object containing folder details and permissions to be granted	getFolderPermission.add(newItem)

The response returned will contain these parameters:

Response Element	**Data Type**	**Description**	**Retrieval Code**
status	Status	Status of the web service call.	getStatus()

setObjectPermission

This function will used to set object permissions.

Request Element	**Data Type**	**Description**	**Setting Code**
objectPermission	List of ObjectPermission	Object permission object containing object details and permissions to be granted	getObjectPermission.add(newItem)

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

getFolderList

Retrieves list of folders in repository

Request Element	Data Type	Description	Setting Code
userName	String	Username for which folders list will be retrived	setUserName()

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()
repository	Folder	Repository Folder	getRepository()
myFolder	Folder	User's Folder	getMyFolder()

getObjectList

Retrieves list of objects in repository

Request Element	Data Type	Description	Setting Code
folderId	String	Folder Id for which objects list will be retrived	setFolderId()
userName	String	Username for which objects list will be retrived	setUserName()

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()
objects	List of Object	List of objects for the web service call	getObjects()

addFolder

This function will be used to add new folder

Request Element	Data Type	Description	Setting Code
userName	String	user Name for which folder will be created	setUserName()
folder	Folder	Folder details to be added	setFolder()

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

deleteFolder

This function will be used to delete folder

Request Element	Data Type	Description	Setting Code
userName	String	user Name for which folder will be deleted	setUserName()
folderId	String	Folder id to be deleted	setFolderId()

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

deleteObject

This function will be used to delete object

Request Element	Data Type	Description	Setting Code
userName	String	user Name for which object will be deleted	setUserName()
objectId	String	Object id to be deleted	setObjectId()

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

Repository Management Web Service Elements

FolderPermission

These are the parameters that you need to set in the FolderPermission object:

Attribute Name	Data Type	Description	Setting Code
folderId	String	Folder id for which permission will be granted	setFolderId()
rolePermission	List of RolePermission	Role permission details	getRolePermission.add(newItem)
userPermission	List of UserPermission	User permission details	getUserPermission.add(newItem)

ObjectPermission

These are the parameters that you need to set in the ObjectPermission object:

Attribute Name	Data Type	Description	Setting Code
objectId	String	Object id	setObjectId()
rolePermission	List of RolePermission	Role permission details	getRolePermission.add(newItem)

userPermission	List of UserPermission	User permission details	getUserPermission.add(newItem)

RolePermission

These are the parameters that you need to set in the RolePermission object:

Attribute Name	Data Type	Description	Setting Code
roleName	String	Role Name	setRoleName()
permission	Integer	Permission	setPermission()

UserPermission

These are the parameters that you need to set in the UserPermission object:

Attribute Name	Data Type	Description	Setting Code
userName	String	User Name	setUserName()
permission	Integer	Permission	setPermission()

Folder

These are the parameters that you can get from the Folder object:

Attribute Name	Data Type	Description	Retrieval Code
folderId	String	Folder Id	getFolderId()
folderName	String	Folder Name	getFolderName()
folders	List of Folder	List of folders for the web service call	getFolders()

Object

These are the parameters that you can get from the Object element:

Attribute Name	Data Type	Description	Retrieval Code
objectId	String	Object Id	getObjectId()
objectName	String	Object Name	getObjectName()

objectType	Integer	Object Type	getObjectType()
folderId	String	Folder Id	getFolderId()

4.5 Cube Management Service

This service will be used to retrieve cube and object repository related information.

Web service functions:

Functions	Description
getCubes	Retrieves list of cubes
setCubePermission	This function will used to set cube permissions
getCubeColumns	This function will used to get cube columns
setCubeColumnPermission	This function will used to set cube columns permissions
setCubeDataPermission	This function will used to set cube data permissions
getCubeDataPermission	This function will used to get cube data permissions
deleteCubeDataPermission	This function will used to delete cube data permissions.
rebuildCube	This function will used to rebuild cube
runCubeSchedulerTask getCubeRebuildStatus runPublishingTask getPublishingTaskStatus	This function will used to run cube scheduler task This function will provide cube rebuild process status. This function will be used to run delivery and publishing task. This function will provide publishing task process status.

Cube Web Service Functions

getCubes

This function will retrive list of cubes **OR** cube details if cube id given

Request Element	Data Type	Description	Setting Code
cubeId	String	cube id(optional) to retrieve specific cube details, if not specified cube list will be returned	setCubeId()

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web	getStatus()

		service call.	
cubes	List of Cube	List of cubes or a single cube details if cube id specified in request	getCubes()

setCubePermission

This function will used to set cube permissions

Request Element	Data Type	Description	Setting Code
cubePermission	List of CubePermission	List of cube permissions to be set	getCubePermission(). add(newItem)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

getCubeColumns

This function will used to get cube columns

Request Element	Data Type	Description	Setting Code
cubeId	String	Cube id for which columns will be returned	setCubeId()

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()
cubeId	String	Cube id for which columns are returned	getCubeId()
Dimensions	String	List of dimensions	getDimensions.get(index)

Measures	String	List of measures	getMeasures.get(index)
OtherColumns	String	List of other columns	getOtherColumns.get(index)
CustomDimensions	String	List of Custom Dimensions	getCustomDimensions.get(index)
CustomMeasures	String	List of Custom Measures	getCustomMeasures.get(index)

setCubeColumnsPermission

This function will used to set cube columns permissions

Request Element	Data Type	Description	Setting Code
cubeId	String	Cube id for which columns permissions will be set	setCubeId()
dimensions	List of CubeColumnPermission	List of cube column permissions to be set	getCubeColumnPermission().add(newItem)
measures	List of CubeColumnPermission	List of cube column permissions to be set	getCubeColumnPermission().add(newItem)
otherColumns	List of CubeColumnPermission	List of cube column permissions to be set	getCubeColumnPermission().add(newItem)
customDimensions	List of CubeColumnPermission	List of cube column permissions to be set	getCubeColumnPermission().add(newItem)
customMeasures	List of CubeColumnPermission	List of cube column permissions to be set	getCubeColumnPermission().add(newItem)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

setCubeDataPermission

This function will used to set cube data permissions

Request Element	Data Type	Description	Setting Code
cubeId	String	Cube id for which permissions will be set	setCubeId()
permissions	List of DataAccessPermission	List of data access permissions to be set	getPermissions().add(newItem)

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

getCubeDataPermission

This function will used to get cube data permissions

Request Element	Data Type	Description	Setting Code
cubeId	String	Cube id for which permissions will be retrived	setCubeId()

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()
cubeId	String	Cube id of retrived cube data permissions	getCubeId()
permissions	List of DataAccessPermission	List of permissions for the cube	getPermissions.get(index)

deleteCubeDataPermission

This function will used to delete cube data permissions

Request Element	Data Type	Description	Setting Code
cubeId	String	Cube id for which permissions will be deleted	setCubeId()
dataAccessPermissionIds	List of Long	Data access permission ids which are to be deleted.	setDataAccessPermissionIds()

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

rebuildCube

This function will used to rebuild cube

Request Element	Data Type	Description	Setting Code
cubeId	String	Cube id of the cube to be rebuild	setCubeId()
incrimental	Boolean	To specify whether the cube will be incrimental or not	setIncrimental()

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

runCubeSchedulerTask

This function will used to run cube scheduler task

Request Element	Data Type	Description	Setting Code
taskName	String	Task name to be run as scheduler task	setTaskName()

The response returned will contain these parameters:

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

getCubeRebuildStatus

This function will provide cube rebuild process status.

Response Element	Data Type	Description	Setting Code
cubeId	String	Cube id for which status will be retrived	setCubeId()

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

runPublishingTask

This function will be used to run delivery and publishing task.

Response Element	Data Type	Description	Setting Code
taskName	String	Task name for which task will be published.	setTaskName()

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

getPublishingTaskStatus

This function will provide delivery and publishing task process status.

Response Element	Data Type	Description	Setting Code
taskName	String	Task name for which	setTaskName()

		status will be retrived.	

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

Cube Management Web Service Elements:

CubePermission

These are the parameters that you need to set in the CubePermission object:

Attribute Element	Data Type	Description	Setting Code
cubeId	String	Cube id for which permissions needs to be set	setCubeId()
roleAccessRights	List of RolePermission	Role access rights to be assigned to cube	getRoleAccessRIghts( ).add(newItem)
userAccessRights	List of UserPermission	User access rights to be assigned to cube	getUserAccessRights( ).add(newItem)

RolePermission

These are the parameters that you need to set in the RolePermission object:

Attribute Name	Data Type	Description	Setting Code
roleName	String	Role name for which permissions will be granted	setRoleName()
read	Boolean	Whether this role will have read permission or not (i.e. True or False)	setRead()

UserPermission

These are the parameters that you need to set in the UserPermission object:

Attribute Name	Data Type	Description	Setting Code
userName	String	User Name for which permissions will be	setUserName()

		granted	
read	Boolean	Whether this user will have read permission or not (i.e. True or False)	setRead()

CubeColumnPermission

These are the parameters that you need to set in the CubeColumnPermission object

Attribute Name	Data Type	Description	Setting Code
columnName	String	Column name for which permissions needs to be set	setColumnName()
roles	List of RolePermission	Roles to be assigned to cube	getRoles().add(newItem)
users	List of UserPermission	Users to be assigned to cube	getUsers().add(newItem)

RolePermission

These are the parameters that you need to set in the RolePermission object:

Attribute Name	Data Type	Description	Setting Code
roleName	String	Role name for which permissions will be granted	setRoleName()
read	Boolean	Whether this role will have read permission or not (i.e. True or False)	setRead()

UserPermission

These are the parameters that you need to set in the UserPermission object:

Attribute Name	Data Type	Description	Setting Code
userName	String	User Name for which permissions will be granted	setUserName()

read	Boolean	Whether this user will have read permission or not (i.e. True or False)	setRead()

DataAccessPermission

These are the parameters that you need to set in the DataAccessPermission object:

Attribute Name	Data Type	Description	Setting Code
dataAccessPermissionId	Long	Data access permission id to set data access permissions	setDataAccessPermissionId()
condition	Condition	Condition contains column name and its value's list	getCondition().add(newItem)
roles	List of RolePermission	Roles to be assigned to cube	getRoles().add(newItem)
users	List of UserPermission	Users to be assigned to cube	getUsers().add(newItem)

Condition

These are the parameters that you need to set in the Condition object:

Attribute Name	Data Type	Description	Setting Code
columnName	String	Column name of the condition	setColumnName()
values	String	List of values for the column	getValues.add(newItem)

Status

These are the parameters that you can get from the Status object:

Attribute Name	Data Type	Description	Retrieval Code
statusCode	String	Status of the web service call. Possible Values : - Success - Fail	getStatusCode()

		- Partial	
detail	List of Detail	Error details of the web service call	getDetail.get(index)

Detail

These are the parameters that you can get from the Detail object:

Attribute Name	Data Type	Description	Retrieval Code
code	String	Error Code of the web service call	getCode()
description	String	Error Description of the web service call	getDescription()
ref	String	Error reference for the web service call	getRef()

Error code and description details

Code	Description	Detail
0	Internal-Server-Error	This error message indicates execution error on server due to the reason specified in ref element under Detail Element
1	Invalid-Input-Data	This error message indicates an invalid input supplied in the request

4.6 Tenant Management Service

This service will be used to create or manage Tenants from third party application to Smarten.

Web service functions:

Functions	Description
addTenant	This function will create new tenant
updateTenant	This function will modify tenant details
deleteTenant	This function will delete tenant
getTenant	This function will get single tenant based on tenant id

Functions	Description
getAllTenant	This function will get list of all tenant

addTenant

This function will create new tenant.

Request Element	Data Type	Description	Code
tenantInfo	TenantInfo	The TenantInfo object holding all of the new Tenant's details	requestObject.getTenantInfo.add(TenantInfo)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

updateTenant

This function will update tenant details.

Request Element	Data Type	Description	Setting Code
tenantInfo	TenantInfo	The TenantInfo object holding all of the role's details	requestObject.getTenantInfo.add(TenantInfo)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

deleteTenant

This function will delete tenant.

Request Element	Data Type	Description	Setting Code
tenantId	String	Tenant Id of the	setTenantId()

		Tenant to be deleted	

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

getTenant

This function will get single tenant based on tenant id.

Request Element	Data Type	Description	Setting Code
tenantId	tenantInfo	The TenantInfo object holding of the tenant's details	requestObject.setTenantId(tenantId)

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

getAllTenant

This function will get list of all tenant.

Request Element	Data Type	Description	Setting Code
-	-	-	-

The response returned will contain these parameters

Response Element	Data Type	Description	Retrieval Code
status	Status	Status of the web service call.	getStatus()

5 SmartenInsight model integration API

Smarten provides REST API for getting details and predictions from the SmartenInsight models. It enables developers to integrate SmartenInsight models in other applications. Following are the list of APIs available:

- Get model data
- Get prediction for single or multiple records (apply and mass apply)
- Get Interpretation details
- Get model summary details

5.1 Get model data

This API provides model output data of SmartenInsight model. API response will be in JSON format.

URL	http://<SMARTENURL>/API/getModelData
Authentication parameters	“username” and “password” in request header
Request data	{ "predictiveId" : "valid predictive id "}
Response data	{"status": 1, "message": "success", "result": [{ "columnName": "<column name>", "columnsValuesList": [< list of data>]}, { "columnName": "<column name>", "columnsValuesList": [< list of data>]}, …..]}
Example	**Request Data :** {"predictiveId" : "p16e7dd732c8.prd"} **Response Data :** { "status": 1, "message": "success", "result": [{ "columnName": "ProductCategory", "columnsValuesList": ["Bakery", "Fruit Juices", "Confectionary", "Cool Drinks", "Alcoholic Drinks",.....] }, { "columnName": "ProductName", "columnsValuesList": ["Cake", "Orange", "Mints", "Soda", "Whisky",....] },

	{"columnName": "ListPrice", "columnsValuesList": [115.68, 171.02, 65.1, 85.9, 310.48,....] }, { "columnName": "SalesPrice", "columnsValuesList": [129.21, 179.42, 77.69, 91.42, 354.29,....] }, { "columnName": "CLUSTER_NUMBER", "columnsValuesList": [2] }, { "columnName": "PREDICTED_ProductCategory", "columnsValuesList": ["Bakery", "Fruit Juices", "Confectionary", "Cool Drinks", "Alcoholic Drinks",....]] }

5.2 Get prediction for single record (Apply)

This API provides facility to get predicted value from model for single record of predictor criteria input. API response is in JSON format.

URL	http://<SMARTENURL>/API/getPrediction
Authentication parameters	“username” and “password” in request header

Request data	{ "predictiveId" : "valid predictive id ", "predictiveData" : [{ "columnName" : "<valid column name>", "columnsValuesList" : <data> }, { "columnName" : " <valid column name> ", "columnsValuesList" : <data> }, ...] }
Response data	{"status": 1, "message": "success", "result": [{ "columnName": "<column name>", "columnsValuesList": [<data>] } , { "columnName": "<column name>", "columnsValuesList": [<data>] } , ...], "accuracy": <accuracy of model> }
Example	**Request Data :** {"predictiveId" : "p16e7dd745c8.prd", "predictiveData" : [{ "columnName" : "ProductName", "columnsValuesList" : ["Cookies"] }, { "columnName" : "SalesQty", "columnsValuesList" : [36.9] }, { "columnName" : "ListPrice",

```
                    "columnsValuesList" : [39.6]
                    },
                    {
                    "columnName" : "SalesPrice",
                    "columnsValuesList" : [69.3]
                    }
                ]
}
```


Response Data


```
{
   "status": 1,
   "message": "success",
   "result": [
      {
         "columnName": "ProductName",
         "columnsValuesList": [  "Cookies"   ]
      },
      {
         "columnName": "SalesQty",
         "columnsValuesList": [   36.9    ]
      },
      {
         "columnName": "ListPrice",
         "columnsValuesList": [   39.6   ]
      },
      {
         "columnName": "SalesPrice",
         "columnsValuesList": [  69.3  ]
      },
      {
         "columnName": "CLUSTER_NUMBER",
         "columnsValuesList": [     2      ]
```

	}, { "columnName": "CLUSTER_LABELS", "columnsValuesList": ["Cluster2"] }], "accuracy": 0.6469589532114715 }

5.3 Get prediction for multiple records (Mass apply)

This API provides facility to get predicted values from model for multiple records of predictor criteria input. API response is in JSON format.

URL	http://<SMARTENURL>/API/getPrediction
Authentication parameters	"username" and "password" in request header
Request data	{ "predictiveId" : "valid predictive id ", "predictiveData" : [{ "columnName" : "<valid column name>", "columnsValuesList" : <list of data> }, { "columnName" : " <valid column name> ", "columnsValuesList" : <list of data> }, ...] }
Response data	{"status": 1, "message": "success", "result": [{ "columnName": "<column name>", "columnsValuesList": [< list of data>]}, { "columnName": "<column name>", "columnsValuesList": [< list of data>] }, ...], "accuracy": <accuracy of model> }
Example	**Request Data :**

	{"predictiveId" : "p16e7dd722c8.prd", "predictiveData" : [{ "columnName" : "EmployeeName", "columnsValuesList" : ["Maude F Setright", "Jason V Mehta", "David Brown"] },{ "columnName" : "ListPrice", "columnsValuesList" : [65.42,219.48,280.48] },{ "columnName" : "SalesPrice", "columnsValuesList" : [78.97,238.25,298.73] }] } **Response Data** { "status": 1, "message": "success", "result": [{ "columnName": "EmployeeName", "columnsValuesList": ["Maude F Setright", "Jason V Mehta", "David Brown"] }, { "columnName": "ListPrice", "columnsValuesList": [65.42, 219.48,

	280.48] }, { "columnName": "SalesPrice", "columnsValuesList": [78.97, 238.25, 298.73] }, { "columnName": "PredictedTarget", "columnsValuesList": [16749.532544518654, 62004.02350336575, 87483.44212763518] }], "accuracy": 0 }

5.4 Get model interpretation

This API provides model interpretation information in HTML format.

URL	http://<SMARTENURL>/API/getInterpretation
Authentication parameters	“username” and “password” in request header
Request data	{ "predictiveId" : "valid predictive id "}
Response data	HTML content of model interpretaion
Example	**Request Data :**

	{"predictiveId" : "p173b881707e.prd"} **Response Data - html content**

5.5 Get model summary information

This API provides model summary information in HTML format.

URL	http://<SMARTENURL>/API/getModelSummary
Authentication parameters	“username” and “password” in request headers
Request data	{ "predictiveId" : "valid predictive id "}
Response data	HTML content of model summary
Example	**Request Data :** {"predictiveId" : "p173c77d4446.prd"} **Response Data - html content**

6 Product and Support Information

Find more information about Smarten and its features at www.smarten.com
Support: support@smarten.com
Sales: sales@smarten.com
Feedback & Suggestions: support@smarten.com
Support & Knowledgebase Portal: support.smarten.com