

**Technical Design Document**

***ODIL Authentication & Session Management***

**OneDealer GmbH**

Real Consulting Group

44 Kifisias Avenue,

151 25 Marousi Attikis

Athens, Greece

***ODIL Authentication & Session Management***

**ThinkRIT Ltd**

1, Persefonis str,

152 34 Gerakas,

Attica, Greece

# ODIL Authentication & Session Management

## Generic ODIL Authentication

In order to be authorized for any ODIL API call, a client has first to create an ODIL session token.

### ODIL - Get session API call

This method returns a session for the requested ODIL api (ApiCode) in conjunction to the provided user credentials.

Get session API call – Authentication token	
Relative URL	//api/sys/Company/AzureLDMS/session/create
Method Type	POST
Content Type	application/json

Arguments			
Header	Authorization	SASToken	REQUIRED
<b>BODY</b>	{ "Target" : { "ApiCode" : "TR:SB:OneDealer:OD" }, "Credentials" : { Username:"", "Password":"" } };		REQUIRED
- <b>APICode</b>	The code of the ODIL API for which access is requested	"TR:SB:OneDealer:OD"	
- <b>Credentials</b>	User/system level credentials		

**Note:** where SASToken must be created with the provided ODIL credentials (e.g User/Key, etc).

For convenience ODIL provides the attached SASTokenGenerator.

## Response

### On Success

```
SuccessResult {  
  BackendCode:string,  
  Session:string,  
  UserClaims:object,    // not used in this implementation  
  UserData:object      // not used in this implementation  
}
```

#### Example

```
{  
  "BackendCode":"5E2D903E-0EC2-47E4-810D-D334A9335E84",  
  "Session":"ed9b8e0c-ab45-436b-adba-de82281f4598",  
  "UserClaims":null,  
  "UserData":null  
}
```

### On Error

```
ServiceError{  
  
  Code                String  
                     System Identifier of error  
  
  type                int  
                     System Identifier of error type Enum  
                     [ communication=0, backendError, authentication,  
                     authorization, session, data, workflow]  
  
  source              Int  
                     System Identifier of error source Enum [ client=0, service ]  
  
  message             String  
                     Human readable error message  
  
  details             String  
                     Human readable error details  
  
  referenceCode       String  
                     Used to find error in logs
```

```

errorData      [
                uniqueltems: false
                ErrorData{
                Key: string,
                Value: string
                }}
}

```

During the Authorization process, if the authentication credentials provided are invalid, the following error occurs

WebException	
<b>StatusCode:Unauthorized</b>	Message: "The remote server returned an error: (401) Unauthorized." Content: <pre> {   "Code": "uri:TR:ErrorCodes:Common:InvalidToken",   "Type": 2,   "Source": 0,   "Message": "Invalid Token",   "Details": null,   "ReferenceCode": "7b44170f-ffb8-458d-8008-ce34a4022b5f",   "ErrorData": null } </pre>
<b>StatusCode:BadRequest</b>	Message: "The remote server returned an error: (400) Bad Request." Content: <pre> {   "Code": "uri:TR:ErrorCodes:AuthenticationFailed",   "Type": 2,   "Source": 0,   "Message": "Authentication failed",   "Details": null,   "ReferenceCode": "edf3804c-3924-4720-a5ed-b59adfd9e019",   "ErrorData": null } </pre>

## ODIL External authentication

In order to support the OD requested authentication process, when requesting access to the ODIL external interface the client must provide some extra request headers with the following information which is transferred to One Dealer.

Request Headers		required
<b>EntityOwnerUserID</b>	<code>actionContext.Request.Headers.TryGetValues("EntityOwnerUserID", out IEnumerable&lt;string&gt; entityOwnerUserIDs);</code>	Yes
<b>EntityOwnerCompanyID</b>	<code>actionContext.Request.Headers.TryGetValues("EntityOwnerCompanyID", out IEnumerable&lt;string&gt; entityOwnerCompanyIDs);</code>	Yes
<b>DimensionCompany</b>	<code>actionContext.Request.Headers.TryGetValues("DimensionCompany", out IEnumerable&lt;string&gt; dimensionCompanies);</code>	No
<b>DimensionBranch</b>	<code>actionContext.Request.Headers.TryGetValues("DimensionBranch", out IEnumerable&lt;string&gt; dimensionBranches);</code>	No
<b>DimensionLocation</b>	<code>actionContext.Request.Headers.TryGetValues("DimensionLocation", out IEnumerable&lt;string&gt; dimensionLocations);</code>	No
<b>DimensionMake</b>	<code>actionContext.Request.Headers.TryGetValues("DimensionMake", out IEnumerable&lt;string&gt; dimensionMakes);</code>	No
<b>DimensionMarketSegment</b>	<code>actionContext.Request.Headers.TryGetValues("DimensionMarketSegment", out IEnumerable&lt;string&gt; dimensionMarketSegments);</code>	No

## Generic Session Management

### ODIL calls and session related errors

After successful session token acquisition, the following steps must be followed for any ODIL API call:

1. Provide the session token in the corresponding request session header (Authorization)
2. Take the renewed session token from the response session header for subsequent use
3. In case of session related errors, the following errors will be returned

Errors	
<b>For session related exceptions:</b>	WebException StatusCode:BadRequest Message: "The remote server returned an error: (400) Bad Request."
<b>For SessionExpired:</b>	Content: { "Code": "uri:TR:ErrorCodes:Common:SessionExpired", "Type": null, "Source": 0, "Message": "Session Expired", "Details": null, "ReferenceCode": "27d10c33-9755-48e5-839f-0fae5d63277b", "ErrorData": null }
<b>For SessionInvalid:</b>	Content: {

	<pre>"Code": "uri:TR:ErrorCodes:Common:SessionInvalid", &gt;Type": null, &gt;Source": 0, &gt;Message": "Invalid Session", &gt;Details": null, &gt;ReferenceCode": "47913d03-587b-40da-b2bd-a066945dccad", &gt;ErrorData": null }</pre>
<b>For unknown exception:</b>	<pre>WebException StatusCode:InternalServerError Message: "The remote server returned an error: (500) Internal Server Error." Content: {   "Code": "uri:TR:ErrorCodes:Common:UnexpectedException",   "Type": null,   "Source": 1,   "Message": "Unexpected Exception of type 'TR.Common.Clients.Http.HttpRequestException' thrown while processing request.",   "Details": "Exception Message was \"The remote server returned an error: (500) Internal Server Error.\". InnerMostException Message was \"The remote server returned an error: (500) Internal Server Error.\"\"",   "ReferenceCode": "72cf665f-f9e8-4e97-832a-9ce0ca687825",   "ErrorData": [     {       "Key": "uri:TR:ErrorData:Common:TraceExceptionString",       "Value": "..."     }   ] }</pre>

# Workflows/Diagrams

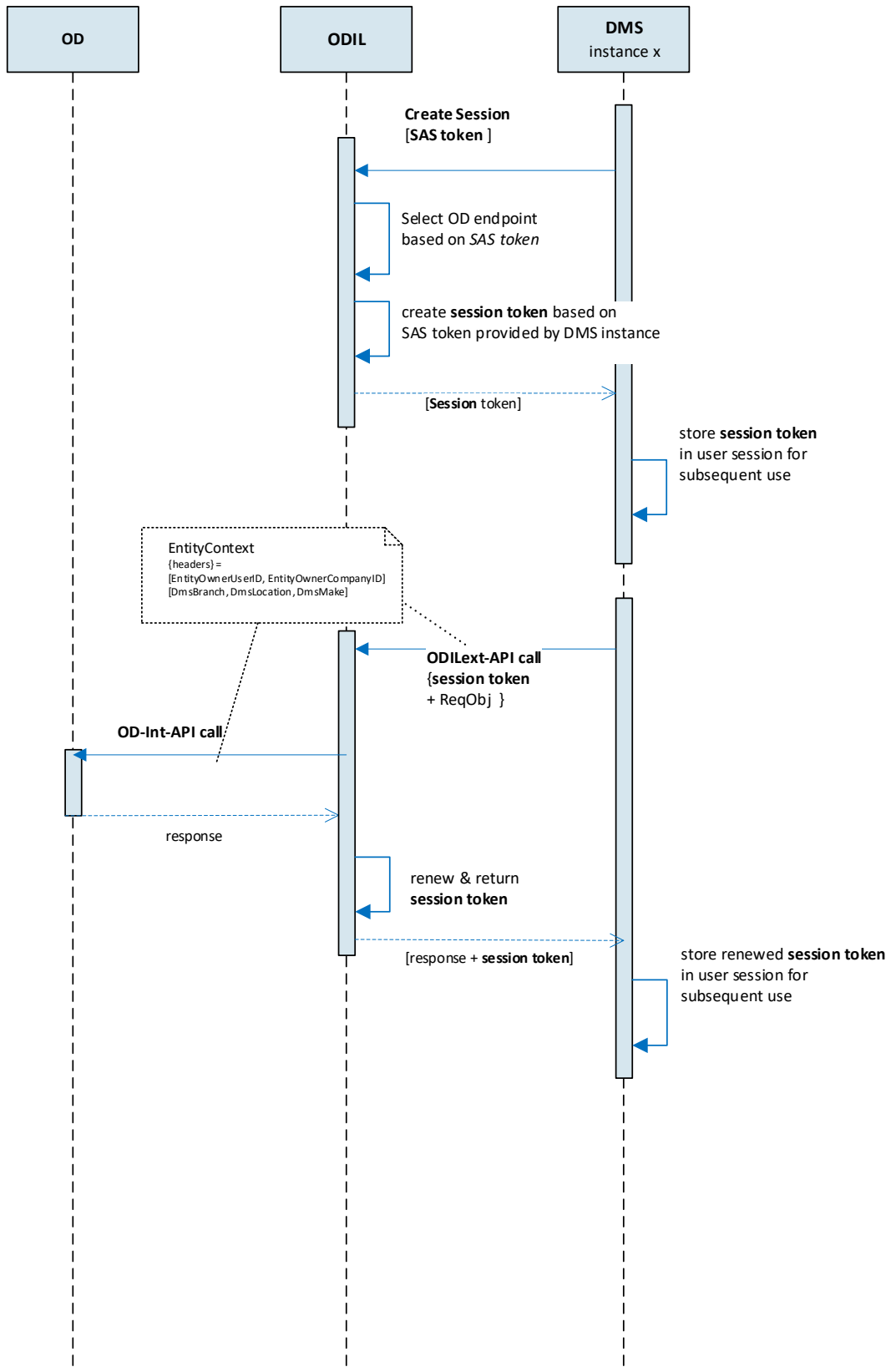


FIGURE 1: ODIL Ext-->Int authentication

## Example

Example: Create Session Token and Use for API invocation

### Example

<https://neu-onedealer-acs-dev.azurewebsites.net/api/sys/Company/AzureILDMS/session/create>

Use the following body:

```
var body = new
{
    Target = new
    {
        ApiCode = "TR:SB:OneDealer:OD"
    },
    Credentials = new { Username="", Password="" }
};
```

with header Authorization = SASToken

where SASToken created with **ILInternalUserKey**

e.g. **ILInternalUserKey** / 2r5u8x/A?D(G+KbPeShVmYq3s6v9y\$B&E)H@McQfTjWnZr4u7x!z%CF-JaNdRgUkXp2s5v8y/B?D(G+KbPeShVmYq3t6w9z\$C&F)H@McQfTjWnZr4u7x!A%D\*G-KaNd

Response is a session token.

Use this for subsequent calls to internal API:

e.g. for CreateOrUpdateMappings

POST

<https://neu-onedealer-integrationlayer-dev.azurewebsites.net/onedealer/api/internal/mappings/sourceEntityCode/targetTypeCode/targetEntityCode/mappings>

where

sourceEntityCode = FuelType,  
targetTypeCode = Company:AzureILDMS,  
targetEntityCode = FuelType

**with request header:**

Session = session

and body:

```
var body = new MappingRequest { Mappings = new Mapping[] { new Mapping { SourceValue = "1", TargetValue = "Diesel" }, new Mapping { SourceValue = "2", TargetValue = "Petrol" } } };
```

**Use the session from response header "**Session**" to reuse the session.**



