

secureID - Passing Secure Parameters



It is possible to pass secure parameters between `javax.servlet.http.HttpServletRequest` and a Cloud session.

Operating principle

This passing of parameters is necessary if the user wishes to launch the application via a `http://host/mywebapp/index.jsp?secureID={RSA}oV8ig8IAFq02y+9reN3b..-type application`.

All the parameters are grouped into an encrypted `<key,value>` map called `secureID`.

Main servlet `index.jsp` structure:

```
<%@taglib prefix="t" tagdir="/WEB-INF/tags" %>

<%@ taglib prefix="adelwagon" uri="http://adelwagon.adelia.hardis.com/adelwagon" %>

<adelwagon:pushCustomVariable key="secureID"><%= (request.getParameter("secureID")!=null ? request.getParameter("secureID") : "")%></adelwagon:pushCustomVariable>

<t:adeliaDesktop title="Adelia Desktop" desktopName="default" frameOptions="SAMEORIGIN">

<jsp:attribute name="head_header">

<adelwagon:customSessionVariables/>

<link href="logo/logo.css" rel="stylesheet" type="text/css"/>

</jsp:attribute>

<jsp:attribute name="head_footer"></jsp:attribute>

<jsp:attribute name="body_header"></jsp:attribute>

<jsp:attribute name="body_footer"></jsp:attribute>

</t:adeliaDesktop>
```

Creating the encrypted `<key,value>` map

[Interactive generation tool](#)

`http://host[:port]/yourapplication/console/cipher.jsp`

Note: you will be asked to log on with a JEE profile that has the "wagon-administrator" role.

Example:

The user wishes to call the page with 3 parameters, the connection profile (login key), the password (password key) and an identifier (id key).

The `secureID` is generated as follows:

tp605.hardis.fr/adelwagon x

tp605.hardis.fr/adelwagonapp/console/cipher.jsp

Clear Text

Encoded Text

encode check

securedID Generator

login=james;
password=mi6;
id=007;

generate securedID

securedID

```
{RSA}ok1WB3/5W5CS2MkAToMhkXJNCN7mdhW+AKMd5h8zaw7kN1cG+x+c8D0zODMcrMQaStNd8q1/bLhGi5oG2Mu7xDmjpeIXYy5A/zIF0TayoaThSbbk8005iXpdCESf+zGSkQRwwNrE796EBzF0gca2MwCNoM7BYi6qsdOfGc3d0iAH5KXPdWxGXhuwLo33Ma5hqtArWxIizukoJCYkuZs2XBLw/KFYI94kR/FEwnKNukxL+LdfWau6wtI89JauVYomz1j3tA+9qKqFkDk5ooWRuPy+UiSzQtIXdaEquCbXdLRSAPQIZHdAtUAVAchUePdUspMVva8bokMJMJmj/FFQ==
```

securedID HTML Encoded

```
%7BRSA%7Dok1WB3%2F5W5CS2MkAToMhkXJNCN7mdhW%2BAKMd5h8zaw7kN1cG%2Bx%2Bc8D0zODMcrMQaStNd8q1%2FbLhGi5oG2Mu7xDmjpeIXYy5A%2FzIF0TayoaThSbbk8005iXpdCESf%2BzGSkQRwwNrE796EBzF0gca2MwCNoM7BYi6qsdOfGc3d0iAH5KXPdWxGXhuwLo33Ma5hqtArWxIizukoJCYkuZs2XBLw%2FKFYI94kR%2FFEwnKNukxL%2BLdfWau6wtI89JauVYomz1j3tA%2B9qKqFkDk5ooWRuPy%2BUiSzQtIXdaEquCbXdLRSAPQIZHdAtUAVAchUePdUspMVva8bokMJMJmj%2FFFQ%3D%3D
```

Generating the securedID via the [servlet /RSAServlet](#)

[http://host\[:port\]/yourapplication/RsaServlet?action=securedID&map=login=james;password=mi6;id=007](http://host[:port]/yourapplication/RsaServlet?action=securedID&map=login=james;password=mi6;id=007)

Calling the page

[http://host\[:port\]/yourapplication/index.jsp?securedID=%7BRSA%7Dok1WB....ePdUspMVva8bokMJMJmj%2FFFQ%3D%3D](http://host[:port]/yourapplication/index.jsp?securedID=%7BRSA%7Dok1WB....ePdUspMVva8bokMJMJmj%2FFFQ%3D%3D)

Note: In the case of a GET, it is advisable to pass the securedID in its encoded URL form.

Retrieving parameters from a Java class

```

secureidMap =null;
String requestSecureID=(String) this.getSessionContext().getMainContainerConfiguration().getAttributes().get("custom.secureID");
if (requestSecureID!=null){  secureidMap = WagonCipher.getInstance().decodeMap(requestSecureID);  if (secureidMap!=null)
{
    String login=secureidMap.get("login");
    String password=secureidMap.get("password");
    String id=secureidMap.get("id");
}
}

```

Note: The *secureID* key is accessible at session attribute level via **custom.secureID**.

Retrieving parameters from a VisualAdelia program

```

ALPHA(256) W_CUSTOMKEY
ALPHA(1024)
W_LOGINALPHA(1024)
W_PASSWORDALPHA(1024) W_IDW_CUSTOMKEY='custom.secureID.login'
APPELER_CLASS 'vatoobx' 'VaToolBxCloudGetSessionAttribute' W_CUSTOMKEY W_LOGIN 1024

W_CUSTOMKEY='custom.secureID.password'
APPELER_CLASS 'vatoobx' 'VaToolBxCloudGetSessionAttribute' W_CUSTOMKEY W_PASSWORD 1024
W_CUSTOMKEY='custom.secureID.id'
APPELER_CLASS 'vatoobx' 'VaToolBxCloudGetSessionAttribute' W_CUSTOMKEY W_ID 1024

```

Retrieving parameters from a LoginModule.

The **secureID** parameter is known by Adelia Cloud. Therefore, it enables "transparent" authentication when the **login** and **password** keys are generated in the *secureID*.

When calling a LoginModule or LoginModule string, all the <key,value> strings are transmitted to the LoginModule.

Example:

An [AdeliaLoginModule](#) receives in the LstAttrRealm list the three login, password and ID keys as well as the associated values.

Example of integration and call from an external application

PHP example

```
<?php
```

```
$payload = file_get_contents('http://host/yourapplication/RsaServlet?action=secureID&map=login=james;password=mi6;id=007');
```

```
echo '<a href="http://host/yourapplication?secureID=',urlencode($payload)," target="_new">Cliquez ici</a>?>
```

JSP example

This example uses the HttpClient API from the Apache project.

```
<%@ page import="org.apache.http.*,org.apache.http.client.*,org.apache.http.impl.client.*,org.apache.http.client.methods.*,java.net.*,java.io.*,java.lang.*  
*,java.util.*"%>
```

```
<%
```

```
String url="http://host/yourapplication/RsaServlet?action=secureID&map=login=james;password=mi6;id=007";
```

```
HttpClient httpClient = HttpClientBuilder.create().build();
```

```
HttpGet rsaRequest = new HttpGet(url);
```

```
HttpResponse rsaResponse = httpClient.execute(rsaRequest);
```

```
BufferedReader rd = new BufferedReader(new InputStreamReader(rsaResponse.getEntity().getContent()));
```

```
    String line = null;
```

```
String secureID = "";
```

```
while ((line = rd.readLine()) != null) {
```

```
    secureID+=line;
```

```
}
```

```
%>
```

```
<a href="http://host/yourapplication?secureID=<%=URLEncoder.encode(secureID)%>" target="_new">Click here</a>
```

[Top of page](#)