Department Service Integration with e-Pramaan

How to integrate a PHP Application

PHP specific integration details are provided in this document. Read e-Pramaan Departments Integration Document before proceeding. ©CDAC Mumbai
Contents
Revision History .................................................................................................................. 2
Abbreviations ...................................................................................................................... 2
1. Introduction ..................................................................................................................... 3
2. Process Flow .................................................................................................................... 3
   1. Single Sign On .............................................................................................................. 3
      A. Login (SSO) initiated from Department Service .................................................. 3
      B. Login (SSO) Initiated at e-Pramaan ................................................................. 4
   2. Single LogOut (SLO) ............................................................................................... 5
      A. Single LogOut (SLO) Initiated by Department service ........................................ 5
      B. Single LogOut (SLO) Initiated by e-Pramaan ..................................................... 7
3. Integration Kit contents ................................................................................................. 7
4. Steps for integration ....................................................................................................... 8
   A. Copy the folders vendor, resources, logs and session to your PHP project .......... 8
   B. Add the e-Pramaan connector file SamlUsingOneLogin.php to project .......... 8
   C. Modify authentication – “Login using e-Pramaan”............................................. 8
   D. Processing SAML Response .................................................................................. 9
   E. Supporting Single LogOut (SLO) ......................................................................... 10
   F. Processing SLO Response ...................................................................................... 11
   G. Web Service Handlers ........................................................................................... 12
      i. SLO Request Handler ......................................................................................... 12
      ii. ssofail handler .................................................................................................... 13
      iii. logoutfail handler .............................................................................................. 13
   H. Processing OneTimeVerification .......................................................................... 14
   I. Session Management .............................................................................................. 15
   J. Modify settings.php ................................................................................................. 15
   K. Register Department Service at e-Pramaan Service Portal .................................... 16

©Centre for Development of Advanced Computing
Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Reason for Change</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.0 (Draft)</strong></td>
<td>22-05-15</td>
<td>e-Pramaan Integration Team</td>
<td>Review</td>
</tr>
<tr>
<td><strong>1.1</strong></td>
<td>08-08-15</td>
<td>e-Pramaan Integration Team</td>
<td>Review</td>
</tr>
</tbody>
</table>

Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML</td>
<td>Security Assertion Mark-up Language</td>
</tr>
<tr>
<td>SP</td>
<td>Service Provider (Department)</td>
</tr>
<tr>
<td>SSO</td>
<td>Single Sign On</td>
</tr>
<tr>
<td>SLO</td>
<td>Single LogOut</td>
</tr>
</tbody>
</table>
**Intended Audience**
The recommended audience for this document is the enterprising person (system administrator/software developer) responsible for e-Pramaan integration at Department end. This document may be useful for the project manager/Department head to assess the effort required for integration with e-Pramaan.

**Prerequisite**
The integrating person at Department end should be well versed in web application development using PHP framework 5.0 and familiar with the work flow of the Department Service.

**1. Introduction**
This document details the steps involved in integrating Department services developed in PHP with e-Pramaan. It explains the process flow and the step-by-step instructional guidance for integrating PHP web application with e-Pramaan to achieve Single Sign On (SSO) and Single LogOut (SLO) functionality provided by e-Pramaan.

**2. Process Flow**

**1. Single Sign On**
Single Sign On (SSO) is an access control mechanism across multiple independent software systems. This allows user to log in once and gain access to all services, without being prompted for log in again at each of them. e-Pramaan allows the user to initiate SSO either from Department Service or from e-Pramaan portal. Similarly Single LogOut session can also be initiated at Department Service or at e-Pramaan portal.

SAML 2.0 is used for SSO implementation.

**A. Login (SSO) initiated from Department Service**
Steps involved in SSO initiated by Department service are depicted in Figure 2.1.
1. User at Department Service initiates SSO by clicking the option to "Login Using e-Pramaan".
2. Department Service then creates SAML SSO request and forwards the user to e-Pramaan for authentication.
3. User is authenticated by e-Pramaan using Challenge-Response mechanism.
4. User is authenticated successfully on e-Pramaan.
5. The user is redirected back to the initiating Department Service. Since the user has been authenticated at e-Pramaan, the Department Service accepts the user and allows him/her to login.
6. If the user fails to authenticate himself/herself on e-Pramaan, the SAML response returns authentication failure.

**Figure 2.1 : SSO initiated from SP Service**

**B. Login (SSO) Initiated at e-Pramaan**

Steps involved in SSO initiated by user at e-Pramaan portal depicted in Figure 2.2.

1. The user directly comes to e-Pramaan portal and logs in using Login and Password.
2. When the user selects a service, e-Pramaan checks the authentication required by the service.
3. According to the requirement of the service, e-Pramaan uses the challenge-response to complete the authentication.
4. The user completes the authentication by providing the appropriate response.
5. e-Pramaan then initiates an SSO session between e-Pramaan and the selected service.
6. The user is then redirected to the selected Department Service where (s)he will be allowed to log in without entering his(her) credentials again.

![Diagram](image)

**Figure 2.2 : SSO initiated from e-Pramaan**

2. **Single LogOut (SLO)**

   During every user session at e-Pramaan, user may log-in into multiple services. When the user is logged out from one service, the user is logged out from all active services. This is achieved by initiating Single LogOut (SLO) either from Department service or e-Pramaan portal. SLO is triggered when the user selects the option to logout, either from the department Service or from e-Pramaan.

   **A. Single LogOut (SLO) Initiated by Department service**

   SLO is initiated by the Department Service when the user decides to logout at the Department Service. The service will Log Out the user locally (or the service may logout the user locally after receiving the response from e-Pramaan) and then creates the SAML SLO request to e-Pramaan to initiate SLO broadcast. This will ensure that the user will be logged out from every service the user was logged in through SSO session. This is depicted in Figure 2.3. The service has an option not to accept the SLO token.
Figure 2.3 : SLO Initiated from Department Service
B. Single LogOut (SLO) Initiated by e-Pramaan

Single LogOut (SLO) will be triggered from e-Pramaan when the user selects the option to logout from e-Pramaan portal. e-Pramaan portal will then broadcast the LogOut Request to all the services which are logged in through the current active session. On receiving the Single LogOut request (SLO request) may terminate the local session and notify e-Pramaan the LogOut status via logout response message. The scenario is depicted in Figure 2.4.

![Diagram of SLO initiated from e-Pramaan](image)

Figure 2.4 : SLO initiated from e-Pramaan

3. Integration Kit contents

PHP web applications require third party libraries for integrating with e-Pramaan. These will be used for validating, signing and encrypting the messages exchanged between e-Pramaan and the Department Service. All the required libraries are provided in the integration kit as explained in the table below.

<table>
<thead>
<tr>
<th>Serial No</th>
<th>Component</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>vendor folder</td>
<td>Vendor folder contains all the required third party libraries required for e-Pramaan integration.</td>
</tr>
<tr>
<td>2</td>
<td>resources folder</td>
<td>Resources folder contains the configuration file</td>
</tr>
</tbody>
</table>
4. Steps for integration

The steps involved in integrating a PHP application with e-Pramaan are listed below. Sample code for integration is provided in the integration kit.

- Copy the folders named vendor, resources, logs and session to the PHP project home folder.
- Copy the php connector named SamlUsingOneLogin.php to the home folder of your project.
- Provide link “Login using e-Pramaan” on the login page of Department Service.
- Modify OnClick event of “Login using e-Pramaan” link to authenticate using e-Pramaan.
- Modify/Implement logic to consume SSO Token sent by e-Pramaan.
- Modify the logout procedure to direct through e-Pramaan.
- Modify/Implement logic to consume logout response from e-Pramaan.
- Implement a RESTful Webservice using the template provided in the Integration Kit (in the rest folder), to receive SLO broadcast request from e-Pramaan.
- Implement logic for OneTimeVerification as detailed further.

A. Copy the folders vendor, resources, logs and session to your PHP project

These folders are required for integration and the contents are explained in the table 3.1 above.

B. Add the e-Pramaan connector file SamlUsingOneLogin.php to project

This file will be the e-Pramaan connector which will assist in integration by acting as a communication link between the Department service and e-Pramaan. This will assist in SAML request creation, signing and encrypting packets between Service and e-Pramaan to ensure secure communication.

C. Modify authentication –“Login using e-Pramaan”
The authentication mechanism in the service application has to incorporate a link to “Login using e-Pramaan”.

In your application, add a button/link named ‘Login using e-Pramaan’.

Using the e-Pramaan connector to create an authentication request for an event generated by onclick(), and send it to e-Pramaan for authenticating the user.

Sample code for authentication is given below. Refer login.php in the integration kit for complete source code.

### Table 3.2 : Code Sample for Authentication

```php
//Check if login button was clicked
if (isset($_REQUEST['login'])) {
    $url = SamlConnectorComp::getValueOf('ePramaanURL');
    $url .= SamlConnectorComp::getValueOf('SingleSignOnServiceURL');

    //initialize the connector
    $saml_conn = new SamlConnectorComp();

    //Create authentication request
    $Auth_Req_Wrapper = $saml_conn->getSamlAuthnRequest();

    //Log the request ID
    $id = $Auth_Req_Wrapper->getId();
    SamlConnectorComp::log_append("AuthRequestID:".$id);

    //Get the authentication request
    $Auth_Req = $Auth_Req_Wrapper->getEncryptedXml();
    $ServiceId = SamlConnectorComp::getValueOf('Issuer');
    SamlConnectorComp::redirect_post($url, $ServiceId, $Auth_Req);
}
```

### D. Processing SAML Response

After successful authentication at e-Pramaan the user is redirected to the Department Service for which SAML request was initiated. This will be a HTTP POST REDIRECT and the user authentication token (SSOToken) will be provided by e-Pramaan in the SAML Response. The service has to consume the SAML Response and decide the further logic for allowing the user to access desired service.

The SAML Response sent by e-Pramaan will be received by a consumer at the Department service end. Consumer will be decode and process the Response before allowing the user access to the service. Sample code for decoding/processing of the SAML Response is given ahead. Refer consume.php in the integration kit for complete source implementation.
Table 3.3 : Code sample for SSO response processing

```php
//Get the Response from HTTP POST
$samlResponse = $_POST['SAMLResponse'];
//Initialize the connector
$saml_conn = new SamlConnectorComp();
//Decode the SAML Response to get SSOToken object
$ssoTokenObject = $saml_conn->processSamlResponse($samlResponse);
//Process the SSOToken and decide whether the user can be allowed to login
if(!empty($ssoTokenObject)) {
    $userSpServicedetail = $ssoTokenObject->getUserSPServiceDetail();
    $sessionID = $ssoTokenObject->getSessionId();
    $serviceUserId = $userSpServicedetail->getServiceUserId();
    SamlConnectorComp::log_append("serviceUserId in Consume: ", $serviceUserId);
    //Set the Session variables
    $_SESSION["ServiceUserId"] = $serviceUserId;
    $_SESSION["SessionIndex"] = $ssoTokenObject->getSessionId();
    //Archive SessionIndex in file (required only if SLO is supported)
    SamlConnectorComp::session_append($sessionID);
    //Redirect to user's home page
    header("Location: home.php");
} else {
    header("Location: error.php");
}
```

E. Supporting Single LogOut (SLO)

A user logged in through e-Pramaan will be able to log-out using Single LogOut (SLO) feature, which implies SLO request initiated will log out user from all the active services for that user.

Whenever a user initiates a logout at service, a SAML SLO request will be sent to e-Pramaan by the Service. The service will have to provide the "Logout" button to incorporate this feature. Sample code for creating SAML SLO request is given in Table 3.4. Refer Home.php in the integration kit for complete source code.

Table 3.4 : Code sample for creating SLO request

```php
//Check if Logout button is clicked
if (isset($_REQUEST['logout'])) {
    //Get e-Pramaan logout request url
    $url = SamlConnectorComp::get_value_of('ePramaanURL');
    $url .= SamlConnectorComp::get_value_of('SingleLogoutServiceURL');
    //Get the current session index
    $session_index = $_SESSION["SessionIndex"];  
    //Log the session index & SLO URL
    SamlConnectorComp::log_append("SessionIndex Logout Req: ", $session_index);
```
F. Processing SLO Response

SLO Response will be sent by e-Pramaan to the initiating Department service in response to the SLO Request. It contains the status whether SLO was success/failure at e-Pramaan. This SLO Response will be sent via "POST Redirect" to the initiating Department service.

- To consume the SLO Response, the Service has to implement a listener to receive and process the SLO Response.
- The SLO Response will have a status *success* if the logout was successful & status as *Request failed* if the logout was not successful at e-Pramaan.
- This listener will process SLO Response & alert the user whether the LogOut operation was successful at e-Pramaan or not.

Sample code for creating Logout Response is given in Table 3.5. Refer `consumeLogoutResponse.php` in the Integration kit for complete source code.

Table 3.5 : Code sample for processing SLO Response

```
//Get the SAML Repsonse
$samlResponse = $_POST['SAMLResponse'];
$saml_conn = new SamlConnectorComp();
//Get Logotut Response
$LogoutResponse = $saml_conn->processLogoutResponse($samlResponse);
//Get Logotut Status from response
$logout_status = $LogoutResponse->getStatus();
SamlConnectorComp::log_append("logout_status": "$logout_status");
//Get session index
```
G. Web Service Handlers

i. SLO Request Handler

e-Pramaan allows user to log in to multiple services through e-Pramaan in a single user session. Suppose that the user is logging out at the e-Pramaan portal, the user should logout from all the associated services for in case of Single LogOut. This is called e-Pramaan initiated Single Logout Service (SLO). e-Pramaan initiated SLO is implemented using RESTful web services.

In e-Pramaan initiated SLO, Department Service will receive a Logout Request from e-Pramaan at the designated REST web service, which then process and validate the request. On successful processing, the service has to logout the user and terminate user session at Department Service. After this, the service has to send the status of the logout in a synchronous REST service response.

Sample code for "SLO Request consumer" WebService is given in Table 3.6.

Table 3.6 : Code sample for receiving SLO Broadcast request

```
//Get the SLO Broadcast Request
$request = file_get_contents("php://input");

//Get ready with encryption seed and salt
$seed = SamlConnectorComp::get_value_of('EncryptionSeed');
$salt = SamlConnectorComp::get_value_of('EncryptionSalt');

//Decrypt & verify the SLO Request
$decrypted_xml = SamlConnectorComp::aes_decrypt($request,$seed, $salt);
SamlConnectorComp::log_append($decrypted_xml);

//Get the validated logout Request
$saml_conn = new SamlConnectorComp();
$SLOLogoutRequest = $saml_conn->processSLOLogoutRequest($decrypted_xml);

//Get Session Index from the SLO Request
$xmlDocument = new DOMDocument();
$xmlDocument->loadXML($SLOLogoutRequest);
$sessionIndex = ";
$tag_name = 'SessionIndex';
$sess_id = $xmlDocument->getElementsByTagName($tag_name);
foreach ($sess_id as $sign) {
    $sessionIndex = $sign->nodeValue;
    SamlConnectorComp::log_append("SessionIndex Fetched:".$sessionIndex);
}
//Clear the local Session Variable & saved Session
$_SESSION = "";
SamlConnectorComp::session_clear($sessionIndex);

$statusCode = "";
//------------------Check whether session is deleted------------------
if (strpos(file_get_contents(dirname(__FILE__)."/Session/sessions.txt"), $sessionIndex) == false) {
    SamlConnectorComp::log_append("SessionIndex Cleared:");
    $statusCode = OneLogin_Saml2_Constants::STATUS_SUCCESS;
} else {
    $statusCode = OneLogin_Saml2_Constants::STATUS_REQUESTER;
}
SamlConnectorComp::log_append("Status Code":PHP_EOL.$statusCode);

//Create SLO Repsonse with appropriate status
$sloLogoutResponse = $saml_conn->createSLOLogoutResponse($sloLogoutRequest, $statusCode);

//Send SLO Repsonse to e-Pramaan
echo $sloLogoutResponse;

ii. ssofail handler
When a user lands at Department Service and tries to log-in using e-Pramaan but fails to get authenticated, a SAML Response is initiated by e-Pramaan. This saml Response will be send via RESTful Web service to the department service.

The service has to implement a RESTful web service for consuming this response. Information provided in the response may be used by the Department Service for logging/auditing purpose. Sample code for this is given in Table 3.7. Refer restssso/index.php in the integration kit for the complete source code.

Table 3.7 : Code sample for SSO failure handling

```php
//Get the response received in SSO fail web service
$request = file_get_contents("php://input");
//Log the received message
SamlConnectorComp::log_append("Restsso Failure AuthnResponse:".$request);
```

iii. logoutfail handler
When the user clicks LogOut button at the Service, it redirects the user to e-Pramaan due to implemented SLO facility. When LogOut attempt is successful, the user is redirected back to the
initiating Service. But, when LogOut fails at e-Pramaan, the user is not redirected back. The LogOut failure, in this case, is intimated via RESTful web service named *logoutfail* web service.

The service has to implement the RESTful web service handlers as mentioned above. Sample code is given in Table 3.8. Refer *restslo/index.php* in the integration kit for the complete source code.

### Table 3.8: Code sample for SLO failure handling

```php
//Get the response received in SLO fail web service
$request = file_get_contents("php://input");
//Log the received message
SamlConnectorComp::log_append("Restsso Failure LogoutResponse:", $request);
```

**H. Processing OneTimeVerification**

User mapping between e-Pramaan and Department Service can be done by either of the two methods explained below.

1. **Adhaar Number based** – when the user is uniquely identified at the Department Service using Adhaar Number this mapping will be used. On successful authentication at e-Pramaan, the user will be redirected to the Department Service along with his Adhaar Number. Service then checks whether an existing account for this Adhaar number exists. If found, user is allowed to login & access the service. Otherwise user will have to register at the SP Service before he is allowed to login using e-Pramaan.

2. **Service UserID Seeding at e-Pramaan** - Service UserID is the UserID used at the Department Service, using which the user logs into the Department Service. This user id will be linked with e-Pramaan ID at e-Pramaan.

If at the service end, Aadhaar seeding is not done for all the users of the service, the service will go for the option **SP UserID seeding at e-Pramaan**. If the user is mapped at e-Pramaan using Service UserID Seeding at e-Pramaan, then at the first login attempt the user will be prompted to do the **OneTimeVerification**. On successful completion of OneTimeVerification existing Service UserID will be linked to his e-Pramaan account.

It is mandatory for the service to provide OneTimeVerification URL at the time of adding a service on portal of e-Pramaan. The logic of OneTimeVerification will validate the user for the first time at the Service end, and push the verification status along with the ServiceUserID to e-Pramaan REST WebService and complete the Service UserID – e-Pramaan account mapping.

When a user tries to access a Service for the first time then e-Pramaan HTTP POST redirects the user to Department service’s OneTimeVerification URL with three request parameters. Request parameters are defined in Table 3.9 below:
Once redirected to the service, the service will ask the user to enter his login credentials. The Service will verify these credentials and the status is pushed back to ePramaan via WebService. The code for this OneTimeVerification push back is provided in OneTimeVerification/OneTimeVerification.cs.

I. Session Management

We have created a file for session management, which is given in the Integration kit (Session/sessions.txt). Kindly refer AuthnResponse/consume.php for code sample on session handling.

J. Modify settings.php

Settings.php is the key configuration file used by the SAML Connector for integrating with e-Pramaan. The entries in the settings.php file are explained as given in Table 3.10 below.

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>Parameters in settings.php</th>
<th>Expected Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1*</td>
<td>ePramaanURL</td>
<td><a href="https://up.epramaan.in">https://up.epramaan.in</a></td>
<td>e-Pramaan specific</td>
</tr>
<tr>
<td>2*</td>
<td>SingleLogoutServiceURL</td>
<td>/processSLORequest.do</td>
<td>e-Pramaan specific</td>
</tr>
<tr>
<td>3*</td>
<td>SingleSignOnServiceURL</td>
<td>/processSSORequest.do</td>
<td>e-Pramaan specific</td>
</tr>
<tr>
<td>4</td>
<td>AssertionIssuer</td>
<td>e-Pramaan</td>
<td>For validation. This value will be string compared to validate the issuer is e-Pramaan</td>
</tr>
<tr>
<td>5</td>
<td>Issuer</td>
<td>ServiceId (Numeric value) given to the service by e-Pramaan when the service is registered at sp.epramaan.in</td>
<td>Every service registered with e-Pramaan will be given a unique id called ServiceId.</td>
</tr>
<tr>
<td>6</td>
<td>SPServiceHomePageURL</td>
<td>SP has to enter this value</td>
<td>Base URL of the SP. Eg: <a href="http://dummyssp.in/localdemo1">http://dummyssp.in/localdemo1</a></td>
</tr>
<tr>
<td>7</td>
<td>SPAssertionConsumerServiceURL</td>
<td>/consume.php</td>
<td>Page which will receive SSO Response from e-Pramaan</td>
</tr>
<tr>
<td>8</td>
<td>SPLLogoutConsumerURL</td>
<td>/consumeLogoutResponse.php</td>
<td>Page which will receive LogoutResponse from e-Pramaan</td>
</tr>
</tbody>
</table>
10* EncryptionSeed * This value will be shared with you by e-Pramaan
e-Pramaan mandates all request and responses be encrypted. This value will be given by e-Pramaan

11* EncryptionSalt *Service ID of the service
This is the Serviceld of the service. Refer item (5) above.

12 SamlSigning True To enable/disable signing of SAML Request.

13 SPCertificateFilePath Path to the private certificate of SP
Private certificate of SP will be used for signing the SAML Request.

14 SPCertificatePassword The private key of SP certificate is protected using this.
This password is required for signing the SAML request/responses using SP’s private key.

* Marked values will be shared to the service by e-Pramaan administrator

K. Register Department Service at e-Pramaan Service Portal
For integration with e-Pramaan, the URLs at Department Service have to be registered with e-Pramaan. The steps for service registration are as follows.

- Department’s Admin registers the department on Service Portal of e-Pramaan.
- e-Pramaan admin gets notification of new SP registered on e-Pramaan.
- After receiving the request through proper channel, e-Pramaan admin activates the registered Department.
- SP admin log in to e-Pramaan and adds new service. While adding service, he enters the required URLs according to Table 3.11 given below.
- e-Pramaan admin activates the newly registered service after testing of integration is completed.
- Now, the Department Service is integrated with e-Pramaan & users can login using e-Pramaan to access the service

Table 3.11 : URL Mapping at e-Pramaan

<table>
<thead>
<tr>
<th>Sr. #</th>
<th>Field name at sp.epramaan.in</th>
<th>Value at SP portal</th>
<th>Value in settings.php</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>SLOURL</td>
<td>/SLOrestService.svc/ConsumeLogoutRequest</td>
<td>URL of the SLO RESTful Web service. Used to validate SLO Request from e-Pramaan.</td>
</tr>
<tr>
<td>10*</td>
<td>EncryptionSeed</td>
<td>* This value will be shared with you by e-Pramaan</td>
<td>e-Pramaan mandates all request and responses be encrypted. This value will be given by e-Pramaan</td>
</tr>
<tr>
<td>11*</td>
<td>EncryptionSalt</td>
<td>*Service ID of the service</td>
<td>This is the Serviceld of the service. Refer item (5) above.</td>
</tr>
<tr>
<td>12</td>
<td>SamlSigning</td>
<td>True</td>
<td>To enable/disable signing of SAML Request.</td>
</tr>
<tr>
<td>13</td>
<td>SPCertificateFilePath</td>
<td>Path to the private certificate of SP</td>
<td>Private certificate of SP will be used for signing the SAML Request.</td>
</tr>
<tr>
<td>14</td>
<td>SPCertificatePassword</td>
<td>The private key of SP certificate is protected using this.</td>
<td>This password is required for signing the SAML request/responses using SP’s private key.</td>
</tr>
<tr>
<td></td>
<td>Service URL</td>
<td>Details</td>
<td>URL</td>
</tr>
<tr>
<td>---</td>
<td>-------------</td>
<td>---------</td>
<td>-----</td>
</tr>
<tr>
<td>1</td>
<td>Service URL</td>
<td>Home page of SP Ex: (<a href="http://dummysp.in/local">http://dummysp.in/local</a> demol)</td>
<td>[SPServiceHomePageURL]</td>
</tr>
<tr>
<td>2</td>
<td>Logout URL</td>
<td>URL of the Servlet at SP which consumes logout response sent by e-Pramaan Ex: (<a href="http://dummysp.in/local">http://dummysp.in/local</a> demol/logoutresponseconsumer)</td>
<td>[SPServiceHomePageURL]+[SPLogoutConsumerURL]</td>
</tr>
<tr>
<td>3</td>
<td>SSO URL</td>
<td>URL of the Servlet at SP which consumes logout response sent by e-Pramaan Ex: (<a href="http://dummysp.in/local">http://dummysp.in/local</a> demol/ssoresponseconsumer)</td>
<td>[SPServiceHomePageURL]+[SPAssertionConsumerService URL]</td>
</tr>
<tr>
<td>4</td>
<td>SLO URL</td>
<td>URL of the REST Web Service at SP which consumes SLO request sent by e-Pramaan Ex: (<a href="http://dummysp.in/local">http://dummysp.in/local</a> demol/ws/saml/SLO)</td>
<td>Not mapped</td>
</tr>
<tr>
<td>5</td>
<td>Logout Failure URL</td>
<td>URL of the REST Web Service at SP Which will receive Logout Response if Logout operation failed at e-Pramaan</td>
<td>Not mapped</td>
</tr>
<tr>
<td>6</td>
<td>SSO Failure URL</td>
<td>URL of the REST Web Service at SP Which will receive SAML Response if SSO operation failed at e-Pramaan</td>
<td>Not mapped</td>
</tr>
<tr>
<td>7</td>
<td>One-time verification URL</td>
<td>The one time verification URL must validate the user at SP Service end and push the verification status to e-Pramaan REST WebService to complete the e-Pramaan User → Service User Id mapping.</td>
<td>Not Mapped</td>
</tr>
</tbody>
</table>
Contact us

Centre for Development of Advanced Computing (C-DAC), Mumbai,
Gulmohar Crossroad 9, Juhu, Mumbai
Tel: +91-22-2620-1606/1574
Fax: +91-22-26232195/ 26210139
Email : epramaan@cdac.in