Use this service to assign a unique authorization id.
This service is used to assign a unique authorization id ('aid').

- When using the Authentication service the library provides patron credentials.
- These credentials are used to look up the patron in either Relais or in the local library system using NCIP LookupUser.
- Once the patron information is retrieved, Relais assigns a unique authorization id ('aid').
- The 'aid' is returned to the library for use in other web services, e.g. passed to Relais via OpenURL.
- Relais uses the 'aid' in all subsequent calls in place of the patron's credentials. The 'aid' is valid only until the patron is idle for the configured period of time.

**Authentication service URL**

http://HOSTNAME/portal-service/user/authentication

HTTP Method: POST

HTTP Header Content-Type: application/json

HTTP Header User-Agent: ${user-agent}

Standard http clients (browser, http client libraries etc.) will automatically include it's User-Agent value.

HTTP Request body: JSON string.

**Input**

Input JSON elements

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApiKey</td>
<td>A library specific API key. Contact <a href="https://help.oclc.org/Resource_Sharing/Relais_D2D/Relais_Discovery_web_services/Authentication">Relais International</a> for your API Key. The API key may be <a href="https://help.oclc.org/Resource_Sharing/Relais_D2D/Relais_Discovery_web_services/Authentication">encrypted</a>. Required</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserGroup</td>
<td>&quot;patron&quot;</td>
</tr>
<tr>
<td></td>
<td>Required.</td>
</tr>
<tr>
<td>PartnershipId</td>
<td>Relais D2D - the Partnership ID indicates to which consortium/group your library belongs.</td>
</tr>
<tr>
<td></td>
<td>Relais ILL - not used.</td>
</tr>
<tr>
<td></td>
<td>Contact <a href="https://help.oclc.org/Resource_Sharing/Relais_D2D/Relais_Discovery_web_services/Authentication">Relais International</a> for your Partnership ID.</td>
</tr>
<tr>
<td>LibrarySymbol</td>
<td>Your Relais library symbol.</td>
</tr>
<tr>
<td></td>
<td>Required.</td>
</tr>
<tr>
<td>PatronId*</td>
<td>The user's patron id or barcode. If using NCIP this is the value that is used to do the NCIP Lookup User. The patron id may be encrypted.</td>
</tr>
<tr>
<td></td>
<td>Caution: For libraries that DO NOT use Relais request management. A dummy patron id is used. Please contact Relais International to set up or obtain the dummy patron id.</td>
</tr>
<tr>
<td>Surname*</td>
<td>The user's surname. The surname may be encrypted.</td>
</tr>
<tr>
<td>UserLogin*</td>
<td>The patron's Relais login. The login may be encrypted.</td>
</tr>
<tr>
<td>UserPassword*</td>
<td>The user's Relais password or if using NCIP a pin that may be required when doing the NCIP Lookup User. The password or pin may be encrypted.</td>
</tr>
</tbody>
</table>

*One or more of these values is required depending on what authentication/authorization is to be done. See examples below.

**Input JSON template**

```json
{
  "ApiKey": "apiKey",
  "UserGroup": "patron",
  "PartnershipId": "partnershipId",
  ...
}
```
"LibrarySymbol": "librarySymbol",
"PatronId": "patronId",
"Surname": "surname",
"UserLogin": "userLogin",
"UserPassword": "userPassword"
}  

**Input json examples for Relais D2D**

**Example 1 - after external authentication - patron's barcode provided for NCIP lookup (encryption used)**

```
{
  "ApiKey": "encrypted API key",
  "UserGroup": "patron",
  "PartnershipId": "test",
  "LibrarySymbol": "OORII",
  "PatronId": "encrypted barcode"
}
```

**Example 2 - after external authentication - patron's barcode and pin provided for NCIP lookup (encryption used)**

```
{
  "ApiKey": "encrypted API key",
  "UserGroup": "patron",
  "PartnershipId": "test",
  "LibrarySymbol": "OORII",
  "PatronId": "encrypted barcode",
  "UserPassword": "encrypted pin"
}
```

**Example 3 - after external authentication - patron's barcode and surname provided for NCIP lookup (encryption used)**

```
{
  "ApiKey": "encrypted API key",
  "UserGroup": "patron",
  "PartnershipId": "test",
  "LibrarySymbol": "OORII",
  "PatronId": "encrypted barcode",
  "Surname": "encrypted surname"
}
```

**Example 4 - after external authentication - patron's barcode and surname provided for NCIP lookup (no encryption used)**

```
"ApiKey": "GYpa21ixF48ssApghf4BFTl7rwUlv4hYauRJ1WAuJfgB9eq30",
"UserGroup": "patron",
"PartnershipId": "test",
"LibrarySymbol": "OORII",
"PatronId": "31883721"
"Surname": "MacKeigan"
}

**Input json examples for Relais ILL**

**Example 1 - after external authentication - patron's barcode provided for NCIP lookup (encryption used)**

```json
{
    "ApiKey": "encrypted API key",
    "UserGroup": "patron",
    "LibrarySymbol": "OORII",
    "PatronId": "encrypted barcode"
}
```

**Example 2 - after external authentication - patron's barcode and surname provided for NCIP lookup (encryption used)**

```json
{
    "ApiKey": "encrypted API key",
    "UserGroup": "patron",
    "LibrarySymbol": "OORII",
    "PatronId": "encrypted barcode",
    "Surname": "encrypted surname",
}
```

**Example 3 - no external authentication - patron's Relais login and password provided for lookup in Relais (encryption used)**

```json
{
    "ApiKey": "encrypted API key",
    "UserGroup": "patron",
    "LibrarySymbol": "OORII",
    "UserLogin": "encrypted Relais login",
    "UserPassword": "encrypted Relais password",
}
```

**Example 4 - no external authentication - patron's Relais login and password provided for lookup in Relais (no encryption used)**

```json
{
    "ApiKey": "GYpa21ixF48ssApghf4BFTl7rwUlv4hYauRJ1WAuJfgB9eq30",
```
"UserGroup": "patron",
"LibrarySymbol": "OORII",
"UserLogin": "mylogin",
"UserPassword": "mypassword",
}

Response

Response elements

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthorizationId</td>
<td>The 'aid' (authorization id) returned by the Authentication service.</td>
</tr>
<tr>
<td>LibrarySymbol</td>
<td>Your Relais library symbol.</td>
</tr>
<tr>
<td>Iso639_2_LangCode</td>
<td>The patron's language as specified in the Relais database (which may be different than the patron's browser language). If no language is specified the default is English.</td>
</tr>
<tr>
<td>FirstName</td>
<td>The patron's first name</td>
</tr>
<tr>
<td>LastName</td>
<td>The patron's surname</td>
</tr>
<tr>
<td>AllowLoanAddRequest</td>
<td>true or false</td>
</tr>
<tr>
<td>AllowCopyAddRequest</td>
<td>true or false</td>
</tr>
<tr>
<td>AllowSelDelivLoanChange</td>
<td>true or false</td>
</tr>
<tr>
<td>AllowSelDelivCopyChange</td>
<td>true or false</td>
</tr>
</tbody>
</table>

Response example

{
  "AuthorizationId": "12knsfd987234ldfd2",
  "LibrarySymbol": "librarySymbol",
  "Iso639_2_LangCode": "ISO639_2_Code",
  "FirstName": "firstName",
  "LastName": "lastName",
}
"AllowLoanAddRequest": true,
"AllowCopyAddRequest": true,
"AllowSelDelivLoanChange": true,
"AllowSelDelivCopyChange": true
}

**Response errors**

This service may return the following error codes and messages:

<table>
<thead>
<tr>
<th>CODE</th>
<th>TYPE</th>
<th>MESSAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUBAN001</td>
<td>Public</td>
<td>Missing parameter</td>
</tr>
<tr>
<td>PUBAN002</td>
<td>Public</td>
<td>Invalid UserGroup</td>
</tr>
<tr>
<td>PUBAN003</td>
<td>Public</td>
<td>Authentication failed. [NCIP_MSG:value]. One of the following reasons may be appended:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• invalid credentials</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• [NCIP_MSG:value]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• invalid cipher text in ${inputParameterName}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• unable to use encrypted ${inputParameterName} value as it is not separated by</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• unable to use encrypted ${inputParameterName} value as one of ApiKey or cipher UTC datetime is missing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• unable to use encrypted ${inputParameterName} value as the cipher UTC datetime (${utcCipherDatetime}) is in wrong format. Supported format for UTC datetime is yyyyMMddHHmmss.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• unable to use encrypted ${inputParameterName} value as cipher UTC datetime (${utcCipherDatetime}) is a future datetime compared with current UTC datetime.</td>
</tr>
<tr>
<td>CODE</td>
<td>TYPE</td>
<td>MESSAGES</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PUBAN004</td>
<td>Public</td>
<td>Service is not available</td>
</tr>
<tr>
<td>PUBAN005</td>
<td>Public</td>
<td>Invalid library symbol</td>
</tr>
<tr>
<td>PUBAN006</td>
<td>Public</td>
<td>ILS server connection timeout error</td>
</tr>
<tr>
<td>PUBAN007</td>
<td>Public</td>
<td>ILS server response timeout error</td>
</tr>
<tr>
<td>PUBAN008</td>
<td>Public</td>
<td>ILS server error</td>
</tr>
<tr>
<td>PUBAN009</td>
<td>Public</td>
<td>Invalid response from ILS server</td>
</tr>
<tr>
<td>PUBAN010</td>
<td>Public</td>
<td>Invalid PartnershipId</td>
</tr>
<tr>
<td>PUBAN011</td>
<td>Public</td>
<td>Invalid aid</td>
</tr>
<tr>
<td>PUBAN012</td>
<td>Public</td>
<td>Invalid API key</td>
</tr>
<tr>
<td>PRIAN001</td>
<td>Private</td>
<td>Internal authentication error occurred. [SERVICE_MSG:Value]</td>
</tr>
<tr>
<td>PRIAN002</td>
<td>Private</td>
<td>Internal error. Error occurred during creating a new patron record</td>
</tr>
</tbody>
</table>

[NCIP_MSG:Value]: This error place holder is available only when the ILS system response includes an error message.

[SERVICE_MSG:Value]: This error places holder is available only when any SQL related error is occurred.

${inputParameterName}: This placeholder is replaced by one of ApiKey, PatronId, Surname, RecordKey, UserLogin or UserPassword.

${utcCipherDatetime}: This placeholder is replaced by the Cipher UTC datetime value.

Note:

[KEY:Value] is an optional error place holder. The UI may need to display the service generated data without translation, like NCIP response’s error message. Therefore, the UI can parse the data easily using the token. Multiple […] may be available, order is not respected.
If the authorization module returns any problem object (error code and message) as response, it will be treated as authentication module's private (internal) error code and message. However the original authorization module's error message will be logged.

**Unsuccessful responses examples**

**Example 1:**

```
{
    "AuthorizationState": {
        "AuthorizationId": "RTExd5ezubHzS2zaS43P3znz6kc1",
        "State": false
    }
}
```

**Example 2:**

```
{
    "ApiKeyOk": false,
    "Problem": {
        "Code": "PUBAZ006",
        "Message": "Invalid API key"
    }
}
```

**Example 3:**

```
{
    "Problem": {
        "Code": "PUBAN001",
        "Message": "Missing required parameter - User login is required."
    }
}
```