

SLZ Application Integration Guide

Table Of Contents

- [Table Of Contents](#)
- [Version History](#)
- [Audience](#)
- [Glossary Of Terms](#)
- [Summary](#)
- [Application Preparation / Prerequisite Steps](#)
- [Login Process / Profile Retrieval](#)
- [Post Login](#)
- [Logout](#)
- [Non-Authenticated Access](#)
- [Appendix A: New SLZ Application Prerequisite Checklist](#)
- [Appendix B: Usage Tracking Infrastructure](#)
- [Appendix C: SLZ API](#)
 - [Auth Interaction](#)
 - [AuthZone redirect](#)
 - [Profile Retrieval](#)
 - [REST Endpoints](#)
 - [Authentication](#)
 - [u.1.0 group-admin](#)
 - [u.2.0 school-admin](#)
 - [u.3.0 teacher](#)
 - [u.4.0 student](#)
 - [u.4.1 student-lexile](#)
 - [c.1.0 school-group](#)
 - [c.2.0 school](#)
 - [c.2.1 school \(full\)](#)
 - [c.3.0 class](#)
 - [c.4.0 student-group](#)
 - [c.5.0 grade-system](#)
 - [c.6.0 school-grade](#)
 - [c.7.0 search](#)

Version History

Version	Date	Editor	Notes
1	2013-03-27	Michael Podrazik < rock@grtmail.com >	<ul style="list-style-type: none">Initial version

2	2013-08-27	Michael Podrazik < rock@grtmail.com >	<ul style="list-style-type: none"> • Add search call • Add version to URLs
---	------------	---	--

Audience

Engineering/Technology staffs responsible for implementing [SLZ applications](#).

Glossary Of Terms

- *AuthToken*: An opaque string passed as a parameter to, or as an element of the path information of, the [AuthZone Redirect URL](#). Identifies the login/access transaction and the user session for [usage tracking](#) purposes.
- *AuthZone*: SLZ's modelling of an application requiring a custom user [profile representation](#), [Redirect URL](#), authorization criteria, etc.
- *JSON*: JavaScript Object Notation is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999 - <http://json.org/>
- *Logout URL*: The URL that SLZ application issues a redirect to after performing any application-specific session cleanup.

```
https://{slzURL}/auth/intl/Logout
```

- *Product*: SLZ's modelling of an application that can be purchased/licensed by customers and made available through SLZ. Often a one-to-one mapping with an [AuthZone](#).
- *Profile*: Representation of SLZ application user's attributes, such as unique identifier, first name, last name, role, etc. Retrieved from SLZ by SLZ application at login.
- *Profile Retrieval URL*: The URL called by SLZ application to retrieve the application-specific profile representation of a user accessing the SLZ application:

```
GET https://{slzURL}/auth/services/Profile/{token}
```

- *redir.html*: An extremely small, static HTML file that must be incorporated into SLZ applications as part of the [usage tracking infrastructure](#).
- *Redirect URL*: URL inside the SLZ application that handles the initial redirect from SLZ on user access. The code behind this URL is responsible for parsing out the [AuthToken](#) and [slzURL](#) and making the [profile retrieval](#) call to SLZ, for example:

```
https://{application-base-url}/{path}?token={token}&slzURL={slzURL}
```

- *REST*: Representational State Transfer. Web service design model - http://en.wikipedia.org/wiki/Representational_state_transfer
- *SLZ*: Scholastic Learning Zone
- *SLZAPI*: [REST](#) API provided to select SLZ applications, primarily centering around retrieval of roster/cohort data of SLZ customers

- *SLZ application*: An application that is integrated with SLZ, in the sense that users login to SLZ to access the application and SLZ serves as the canonical source of profile and roster information for users of the application.
- *slzURL*: URL passed as a parameter to, or as an element of the path information of, the AuthZone redirect URL. Identifies the SLZ instance that the user logged in from, where the profile should be retrieved from, as well as any subsequent API calls.
- *usage.js*: A JavaScript file that must be sourced into all relevant SLZ application pages as part of the [usage tracking infrastructure](#)

Summary

The following defines the Scholastic Learning Zone application integration process and is intended for use as a simple overview for developers of [SLZ applications](#).

The SLZ Platform provides the following services for SLZ applications:

1. Authentication of SLZ users
2. Authorization of authenticated users for a given SLZ application
3. Gateway for users to access SLZ applications
4. Exposure of application-specific representation of authenticated/authorized users' [profile](#) information to SLZ applications
5. Usage and time tracking of users' sessions in SLZ applications
6. Exposure of [API](#) centering around roster/cohort data of customers to select SLZ applications

Minimal expectations are placed on SLZ applications. They are assumed to be web applications capable of:

1. Basic session tracking
2. Issuing of HTTP 302 style browser redirects
3. Parsing of URL parameters
4. Making [REST](#) style HTTP requests and operating on the response

Application Preparation / Prerequisite Steps

(See: [Appendix A: New SLZ Application Prerequisite Checklist](#))

SLZ applications must be registered with SLZ as an [AuthZone](#) and likely a [Product](#). An AuthZone is SLZ's modelling of an application requiring a custom user [profile representation](#), [redirect URL](#), authorization criteria, etc. A Product is SLZ's modelling of an application that can be purchased/licensed by customers and made available through SLZ. There is often a one-to-one mapping of an AuthZone and a Product.

AuthZones are used by the SLZ system when a user logs in and/or accesses a given SLZ application. Products are used by SLZ staff during order input and by customer administrators and teachers when enrolling users.

In order to register as an AuthZone with SLZ, the SLZ application staff must provide a URL that SLZ will redirect to on successful access by users. The application code handling requests to this URL must be prepared to accept two string parameters, the [AuthToken](#) and [slzURL](#). The particular way that these are embedded in the URL is up to the SLZ application. It may be a URL parameter, an element of the path information, etc.

In addition to the redirect URL, the SLZ application staff must provide a list of user attributes the application requires as well as a preferred format for profile interchange. SLZ has great flexibility in this regard, although [JSON](#) is recommended as it is the only format currently supported by the [SLZ API](#).

Lastly, the SLZ application staff must incorporate the [redir.html](#) and [usage.js usage tracking infrastructure](#) components into the SLZ application.

Login Process / Profile Retrieval

The details of how users originally access SLZ is out of scope for this document. Suffice it to say that users know how to access SLZ and have appropriate login credentials to enter the SLZ portal. At that point, they are provided with links to the SLZ applications that they are authorized to access. When a user selects a SLZ application to access, the following occurs:

1. SLZ re-authorizes the user for the SLZ application they are attempting to access
2. SLZ generates an [AuthToken](#). The AuthToken, the appropriate [slzURL](#), the destination [Redirect URL](#) provided by the SLZ application staff and other parameters relevant to the [usage tracking infrastructure](#) are encoded into a URL pointing at the [redir.html](#) file hosted by SLZ application and a browser redirect is issued.
3. The [redir.html](#) page is loaded by the user's browser and JavaScript sourced into that document is executed. This results in certain data being stored in a cookie scoped to the SLZ application's domain. This data is used later by the [usage.js](#) script included on SLZ application pages. When the cookie is set, a browser redirect is issued to the [Redirect URL](#) provided by SLZ application staff.
4. The [Redirect URL](#) is loaded by the user's browser. The [AuthToken](#) and [slzURL](#) parameters are parsed out by SLZ application. SLZ application initiates a HTTP GET request to the [Profile Retrieval URL](#)

```
GET https://{slzURL}/auth/services/Profile/{token}
```

Assuming that the token is valid and there are no networking issues, etc. a response is generated containing the application-specific [profile representation](#):

```
{
  identifier: 'abc123',
  role: 'gadmin|sadmin|teacher|student',
  firstName: 'abc',
  lastName: 'xyz',
  //...
```

}

5. The user is now logged in to SLZ application

The AuthToken has the following additional characteristics that must be kept in mind:

1. It is one-time-use. Subsequent profile retrieval attempts using a token that has already been used will result in a HTTP 400 “Bad Request” response, with a message indicating the token has already been consumed.
2. It is only valid for 60 seconds from the time of generation. A profile retrieval attempt using a token that is older than that will result in a HTTP 400 “Bad Request” response, with a message indicating that the token has expired.

A profile retrieval attempt without a token will result in a HTTP 400 “Bad Request” response, with a message indicating that a token is required.

If an error occurs in processing the profile retrieval request, a HTTP 500 “Internal Server Error” response will be issued, with a message indicating that an error occurred processing the request.

Post Login

At this point, the user has been authenticated and authorized by SLZ, the [profile](#) has been retrieved by the SLZ application and the user can be considered logged in.

It is a requirement that no details of the user profile be saved to persistent storage by SLZ application other than the identifier property of the user due to the mutability of the data. Only the identifier is guaranteed to be unique and unchanging and should thus serve as the key by which application-specific profile data is associated with users. Other profile data such as name, etc. should only be saved in the session and considered transient.

Logout

When the user chooses to logout of the SLZ application, it is expected that the SLZ application will perform whatever cleanup of the user’s session is required and then redirect the user’s browser to the pre-configured SLZ [Logout URL](#):

```
https://{slzURL}/auth/intl/Logout
```

This will result in any session state currently being maintained by SLZ being destroyed and the user’s browser being redirected to the SLZ Login Page.

Non-Authenticated Access

If a request is received by SLZ application from a browser without an established session in SLZ application, the browser should be redirected to the SLZ [Logout URL](#) above.

Appendix A: New SLZ Application Prerequisite Checklist

Performed By	Description
SLZ application staff	Provide redirect URL and requested profile attributes and format to SLZ staff
SLZ staff	Creation of Product and/or AuthZone records.
SLZ staff	Creation of application-specific profile generator.
SLZ staff	Incorporation of Product into relevant order-entry, enrollment, gateway, etc. screens of SLZ portal.
SLZ staff	Provide redir.html and usage.js snippets to SLZ application staff
SLZ staff	Provide application username and password to SLZ application staff for API access, (if applicable)
SLZ application staff	Incorporation of redir.html and usage.js usage tracking infrastructure components into SLZ application

Appendix B: Usage Tracking Infrastructure

From the SLZ application's perspective, the SLZ usage tracking infrastructure consists of the following two items:

1. A single file, [redir.html](#), incorporated into, and made web-accessible from, the SLZ application. The only requirement is that it be served from the same root domain as the rest of the SLZ application. Other path information may be determined by SLZ application staff.
2. A single script element that incorporates the [usage.js](#) tracking script. This element must be included on all pages of SLZ application. The most logical place is in a header or footer section of a common page template used on all SLZ application pages.

Appendix C: SLZ API

Auth Interaction

AuthZone redirect

Post successful user authentication and authorization, Auth will redirect to application's pre-configured location, (for example):

`https://{application-base-url}/{path}?token={token}&slzURL={slzURL}`

1. token should be retained by application for profile retrieval call below
2. slzURL should be retained by application for profile retrieval call below as well as future API access

Profile Retrieval

Profile retrieval call is made to slzURL and includes token parameter from previous AuthZone redirect.

```
GET https://{slzURL}/auth/services/Profile/{token}
{
  uuid: 'abc123',
  role: 'gadmin|sadmin|teacher|student',
  firstName: 'abc',
  lastName: 'xyz'
  // also include all other properties contained in role-specific
  //   user call listed below, for example:
  //   - u.1.0 group-admin
  //   - u.2.0 school-admin
  //   - u.3.0 teacher
  //   - u.4.0 student
}
```

REST Endpoints

Authentication

All REST calls must include a username and password, (previously provided to the calling application prior to integration), via HTTP Basic Authentication headers - http://en.wikipedia.org/wiki/Basic_access_authentication#Protocol

u.1.0 group-admin

```
GET https://{slzURL}/slzapi/v1/group-admins/{group-admin-identifier}
{
  identifier: 'a123',
  firstName: 'jb',
  lastName: 'nm'
  schoolGroup: {
    identifier: 'b456',
    name: 'abc'
  }
}
```

u.2.0 school-admin

```
GET https://{slzURL}/slzapi/v1/school-admins/{school-admin-identifier}
{
  identifier: 'a123',
  firstName: 'jb',
  lastName: 'nm'
  school: {
    identifier: 'b456',
    name: 'abc'
  }
}
```

```
}  
}
```

u.3.0 teacher

GET https://{slzURL}/slzapi/v1/teachers/{teacher-identifier}

```
{  
  identifier: 'a123',  
  firstName: 'jb',  
  lastName: 'nm'  
  school: {  
    identifier: 'b456',  
    name: 'abc'  
  },  
  classes: [ // can be empty  
    {  
      identifier: 'c789',  
      name: 'def'  
    },  
    {  
      identifier: 'd012',  
      name: 'ghi'  
    }  
  ]  
}
```

Notes: *classes will contain all of the classes that the teacher is associated with.*

u.4.0 student

GET https://{slzURL}/slzapi/v1/students/{student-identifier}

```
{  
  identifier: 'a123',  
  firstName: 'jb',  
  lastName: 'nm'  
  school: {  
    identifier: 'b456',  
    name: 'abc'  
  },  
  grade: {  
    scholasticGradeCode: 'K',  
    scholasticGradeName: 'Kindergarten',  
    localGradeCode: 'F',  
    localGradeName: 'Foundation',  
    localGradeSystemCode: 'AUS'  
  },  
  classes: [ // can be empty  
    {  
      identifier: 'c789',  
      name: 'def'  
      studentGroups: [ // can be empty
```



```

        {
            identifier: 'c789',
            name: 'def'
        },
        {
            identifier: 'd012',
            name: 'ghi'
        }
    ]
},
{
    identifier: 'd012',
    name: 'ghi',
    studentGroups: [] // can be empty
}
]
}

```

Notes: *classes will contain all classes the student is a member of. If the student is a member of one or more student-groups in the class, the student-group/s will appear as children of the class. If the student is not a member of any student-groups in the class, then the studentGroups array will be empty. No student-groups will appear that the student is not a member of.*

u.4.1 student-lexile

PUT <https://{slzURL}/slzapi/v1/students/{student-identifier}/lexile>
 {lexileScore} // integer

c.1.0 school-group

GET <https://{slzURL}/slzapi/v1/school-groups/{school-group-identifier}>

```

{
    identifier: 'a123',
    name: 'xyz',
    parentSchoolGroup: { // can be null
        identifier: 'b456',
        name: 'abc'
    },
    childSchoolGroups: [ // can be empty
        {
            identifier: 'c789',
            name: 'def'
        },
        {
            identifier: 'd012',
            name: 'ghi'
        }
    ],
    childSchools: [ // can be empty
        {

```

```

        identifier: 'e345',
        name: 'jkl'
    },
    {
        identifier: 'f678',
        name: 'mno'
    }
]
}

```

c.2.0 school

GET https://{slzURL}/slzapi/v1/schools/{school-identifier}

```

{
    identifier: 'a123',
    name: 'xyz',
    schoolGroups: [ // can be empty
        {
            identifier: 'b456',
            name: 'abc'
        }
    ],
    classes: [ // can be empty
        {
            identifier: 'c789',
            name: 'def'
        },
        {
            identifier: 'd012',
            name: 'ghi'
        }
    ]
}

```

c.2.1 school (full)

GET https://{slzURL}/slzapi/v1/schools/{school-identifier}/full

```

{
    identifier: 'a123',
    name: 'xyz',
    schoolGroups: [ // can be empty
        {
            identifier: 'b456',
            name: 'abc'
        }
    ],
    classes: [ // can be empty
        {
            identifier: 'c789',
            name: 'def'
            studentGroups: [ // can be empty

```

```

{
  identifier: 'c789',
  name: 'def'
  students: [ // can be empty
    {
      identifier: 'e345',
      firstName: 'jkl',
      lastName: 'mno'
    },
    {
      identifier: 'f678',
      firstName: 'pqr',
      lastName: 'stu'
    }
  ]
},
{
  identifier: 'd012',
  name: 'ghi',
  students: []
}
],
students: [ // can be empty
  {
    identifier: 'g901',
    firstName: 'vwx',
    lastName: 'yza'
  },
  {
    identifier: 'h234',
    firstName: 'bcd',
    lastName: 'efg'
  }
],
teachers: [
  {
    identifier: 'e345',
    firstName: 'jkl',
    lastName: 'mno'
  },
  {
    identifier: 'f678',
    firstName: 'pqr',
    lastName: 'stu'
  }
]
},
{
  etc...
}
]

```

```
}
```

Notes: *students will contain all students that are not a member of any student-group in the class. If a student is a member of one or more student-groups in the class, they will appear in the students array associated with the appropriate student-group/s.*

c.3.0 class

GET <https://{slzURL}/slzapi/v1/classes/{class-identifier}>

```
{
  identifier: 'a123',
  name: 'xyz',
  school: {
    identifier: 'b456',
    name: 'abc'
  },
  studentGroups: [ // can be empty
    {
      identifier: 'c789',
      name: 'def'
      students: [ // can be empty
        {
          identifier: 'e345',
          firstName: 'jkl',
          lastName: 'mno'
        },
        {
          identifier: 'f678',
          firstName: 'pqr',
          lastName: 'stu'
        }
      ]
    },
    {
      identifier: 'd012',
      name: 'ghi',
      students: []
    }
  ],
  students: [ // can be empty
    {
      identifier: 'g901',
      firstName: 'vwx',
      lastName: 'yza'
    },
    {
      identifier: 'h234',
      firstName: 'bcd',
      lastName: 'efg'
    }
  ]
}
```

```

],
teachers: [
  {
    identifier: 'e345',
    firstName: 'jkl',
    lastName: 'mno'
  },
  {
    identifier: 'f678',
    firstName: 'pqr',
    lastName: 'stu'
  }
]
}

```

Notes: *students will contain all students that are not a member of any student-group in the class. If a student is a member of one or more student-groups in the class, they will appear in the students array associated with the appropriate student-group/s.*

c.4.0 student-group

GET <https://{slzURL}/slzapi/v1/student-groups/{student-group-identifier}>

```

{
  identifier: 'a123',
  name: 'xyz',
  class: {
    identifier: 'b456',
    name: 'abc'
  },
  students: [ // can be empty
    {
      identifier: 'e345',
      firstName: 'jkl',
      lastName: 'mno'
    },
    {
      identifier: 'f678',
      firstName: 'pqr',
      lastName: 'stu'
    }
  ]
}

```

c.5.0 grade-system

GET <https://{slzURL}/slzapi/v1/grade-systems/{grade-system-code}>

```

{
  code: 'AUS',
  name: 'Australia',
  grades: [

```

```

    {
      scholasticGradeCode: 'PK',
      scholasticGradeName: 'Pre-Kindergarten',
      localGradeCode: 'P',
      localGradeName: 'Preschool'
    },
    {
      scholasticGradeCode: 'K',
      scholasticGradeName: 'Kindergarten',
      localGradeCode: 'F',
      localGradeName: 'Foundation'
    },
    {
      scholasticGradeCode: '1',
      scholasticGradeName: 'First Grade',
      localGradeCode: '1',
      localGradeName: 'Year 1'
    },
    //...
  ]
}

```

c.6.0 school-grade

GET https://{slzURL}/slzapi/v1/schools/{school-identifier}/{scholastic-grade-code}

```

[
  {
    identifier: 'g901',
    firstName: 'vwx',
    lastName: 'yza'
  },
  {
    identifier: 'h234',
    firstName: 'bcd',
    lastName: 'efg'
  }
]

```

c.7.0 search

GET https://{slzURL}/slzapi/v1/search?q={query}

GET https://{slzURL}/slzapi/v1/search?q={query}&user-identifier={user-identifier}&school-identifier={school-identifier}&types=students,classes

Query Parameters:

- q: the search query (*Required*)
- user-identifier: the user identifier (*Optional*)
- school-identifier: the school identifier (*Required for searching "below" school level*)

- `types`: comma-separated list of "types" to search for. *(Optional)*
Possible values:
 - "students"
 - "student-groups"
 - "classes"
 - "schools"
 - "school-groups"

Access Control:

Access control is optional and based on the presence of the user-identifier query parameter. If this parameter is not present, access is presumed to be by the trusted [SLZ application](#) itself.

If the user-identifier parameter is present, [SLZAPI](#) will restrict access to types available to the particular user identified. For example, if the user-identifier belongs to a teacher, they will not be able to search for school-groups or schools; and classes, student-groups and students returned in search results will be limited to those entities that they are associated with in the roster graph.

```
{
  students: [
    {
      identifier: 'g901',
      firstName: 'vwx',
      lastName: 'yza'
    },
    {
      identifier: 'h234',
      firstName: 'bcd',
      lastName: 'efg'
    }
  ],
  classes: [
    {
      identifier: 'a098',
      name: 'm123'
    },
    {
      identifier: 'b765',
      name: 'n432'
    }
  ]
}
```