Service Security Layers

Liferay's remote services are secured by default. They sit behind a layer of security that allows only local connections. To invoke Liferay services from a remote client, you must take deliberate steps to enable such access. Liferay's core web services require user authentication and verification. Regardless of whether you call the remote service from the same machine or via a web service, Liferay's standard security model performs its function. The user invoking a web service must have the proper permissions (as defined by Liferay's permissions system) for the remote service invocation to complete successfully. This tutorial explains these processes.

The first layer of security that a client encounters when calling a remote service is called *invoker IP filtering*. Imagine that you have have a batch job that runs on another machine in your network. This job polls a shared folder on your network and uses Liferay's web services to upload documents to your Liferay site's *Documents and Media* app on a regular basis. To get your batch job through the IP filter, you must grant web service access to the machine the batch job is running on. For example, if your batch job uses Liferay's SOAP web services to upload the documents, you must add the IP address of the machine where the batch job runs to the `axis.servlet.hosts.allowed` property. A typical entry might look like this:

```
axis.servlet.hosts.allowed=192.168.100.100, 127.0.0.1, [SERVER_IP]
```

If the IP address of the machine where the batch job runs is listed as an authorized host for the service, the machine is allowed to connect to Liferay's web services, pass in the appropriate user credentials, and upload the documents.

Note: The `portal.properties` file resides on the Liferay instance's host machine and is controlled by the instance administrator. Instance administrators can configure security settings for the Axis Servlet, the Liferay Tunnel Servlet, the Spring Remoting Servlet, the JSON Servlet, the JSON Web Service Servlet, and the WebDAV Servlet. The `portal.properties` file (online version is available at https://docs.liferay.com/dxp/digital-enterprise/7.0-latest/propertiesdoc/portal.properties.html) describes these properties.

Next, if you invoke the remote service via web services (e.g., JSON WS, old JSON, Axis, REST, etc.), a two step process of authentication and authentication verification takes place. Each call to a Liferay web service must be accompanied by a user authentication token: `p_auth` . It's up to the web service caller to produce the token

enters Liferay's user permission layer.

Liferay's user permission layer is the last Liferay security layer triggered when services are invoked remotely. It's used for every object in the Liferay instance, regardless of whether a local or remote service is involved. The user ID associated with a web service invocation must possess the proper permission to operate on the objects it's trying to access. A remote exception is thrown if the user ID doesn't possess the required permissions. An instance administrator can grant users access to these resources. For example, suppose you created a Documents and Media Library folder called *Documents* in a site, created a role called *Document Uploaders*, and granted this role the rights to add documents to your new folder. If your batch job accesses Liferay's web services to upload documents into the folder, you must call the web service using a user ID of a member of this role (or using the user ID of a user with individual rights to add documents to this folder, such as an instance administrator). If you don't, Liferay denies you access to the web service.

When invoking remote Liferay services from a non-browser client, you can specify the user credentials using HTTP basic authentication. For security reasons, you must be logged in and supply a valid `p_auth` authentication token to invoke a Liferay web service via a browser. Since you should never pass credentials over the network unencrypted, we recommend using HTTPS whenever accessing Liferay services on an untrusted network. Most HTTP clients (e.g., cURL) let you specify the basic authentication credentials in the URL–this is very handy for testing.

Important: To invoke a Liferay web service via your browser, you must be logged in to Liferay. You must also supply an authentication token (the `p_auth` parameter). If you navigate to your Liferay instance's JSON web services API page (localhost:8080/api/jsonws, by default) and click on a remote service method, you'll see the `p_auth` token for your browser session. This token is automatically supplied when you invoke a Liferay web service via the JSON web services API page or via JavaScript using `Liferay.Service(...)`.

Use the following syntax to call the Axis web service using credentials.

```
http://" + emailAddressOrScreenNameOrUserIdAsString + ":" + password + "@[server.com]:
[port]/api/axis/" + serviceName
```

The `emailAddressOrScreenNameOrUserIdAsString` should be the user's email address, screen name, or user ID. The Liferay instance's authentication type setting determines which one to use. A user can find his or her ID by logging in as the user and accessing *My Account* → *Account Settings* from the User Menu. On this interface, the user ID appears below the user's profile picture and above the birthday field.

Suppose that your Liferay instance is set to authenticate by user ID, and that there's a user with an ID of `2` and a password of `test`. You can access Liferay's remote Organization service with the following URL:

Note that if an email address appears in the URL path, it must be URL-encoded (e.g. `test@example.com` becomes `test%40liferay.com`).

Suppose that your Liferay instance is now set to authenticate by email address. To call the same web service for the same user, change the URL to this:

```
http://test%40liferay.com:test@localhost:8080/api/axis/Portal_OrganizationService
```

As mentioned, the authentication type specified for your Liferay instance dictates the authentication type you'll use to access your web service. The instance administrator can set the instance's authentication type to email address, screen name, or user ID.

You can set the authentication type via the Control Panel or via the `portal-ext.properties` file. To set the authentication type via the Control Panel, navigate to *Control Panel* → *Configuration* → *Instance Settings*, and select the *General* tab under *Authentication*. Choose your authentication type in the *How do users authenticate?* menu. To set the authentication type via properties file, add the following lines to your Liferay instance's `portal-ext.properties` file and uncomment the line for the appropriate authentication type:

```
#company.security.auth.type=emailAddress
#company.security.auth.type=screenName
#company.security.auth.type=userId
```

You should also review your Liferay instance's password policies, since they'll be enforced on your administrative user as well. If the instance enforces password policies on its users (e.g., requires them to change their passwords on a periodic basis), an administrative user accessing Liferay's web services in a batch job will have his or her password expire too.

To prevent a password from expiring, an instance administrator can add a new password policy that doesn't enforce password expiration, and then add a specific administrative user to the policy. Then your batch job can run as many times as you need it to, without your administrative user's password expiring.

To summarize, accessing Liferay remotely requires you to pass the following layers of security checks:

- *IP permission layer*: The IP address must be pre-configured in the server's portal properties.
- *Authentication/verification layer (web services only)*: Liferay verifies that the caller's authorization token can be associated with an instance user.
- *User permission layer*: The user needs permission to access the related resources.

Related Topics

Configuring JSON Web Services

Invoking Remote Services

Invoking JSON Web Services

JSON Web Services Invoker

JSON Web Services Invocation Examples

SOAP Web Services

« Invoking Remote Services

Registering JSON Web Services »

¿Fue útil este artículo?

Sí No

Usuarios a los que les pareció útil: 0 de 0

¿Necesitas más ayuda?

Abre un ticket

Contacta con nosotros

Pregunta en la Comunidad

Liferay University

Forma a tu equipo en los puntos esenciales del uso de Liferay con videos detallados y ejercicios prácticos diseñados para simular una formación presencial

Liferay Global Services

Aumenta tu conocimiento sobre el producto y el desarrollo gracias a expertos que te ayudarán a maximizar tu ROI

© 2023 LIFERAY INC. ALL RIGHTS RESERVED