

File System

WebGLStudio allows to store files server side so all your changes can be persistent between sessions even if you login from different computers.

It also enables to share your creations between other users or reuse components/prefabs created by other users.

To achieve this the WebGLStudio uses the library [LiteFileSystem](#).

DriveModule

The Module in the system in charge of all the resources is called DriveModule.

Here is an example of how to save a resource in the system:

```
var res = new LS.Resource();
res.filename = "guest/temp/foo.json";
res.data = JSON.stringify({ name: "javi" });
```

```
DriveModule.saveResource( res );
```

litefileserver.js

LiteFileServer.js is front-end and back-end library that allows javascript apps to store resources (images, text-files, binaries) in the server. It comes with its own users and units system that allow every user to partition its own space and share it among other users.

Some of the features:

- REST HTTP API for storing, listing, moving, updating or deleting files.
- Basic users (register, login, delete, administration)
- Independent file tree per user
- Units, users can have several units to store files and share with other users
- Files can have thumbnail image and metadata

The server side is coded in PHP and comes with useful scripts for administration.

Installing

Check the [install guide](#) to see a step by step process of how to install it in your server.

Usage

Once installed you can include the `litefileserver.js` script in your project you must first login:

```
var lfs = LFS.setup("myhost", onReady );
var session = null;

//check to see if the server is available
function onReady()
{
    LFS.login( username, password, onLogin );
}

function onLogin( my_session, err )
{
    if(!session)
        throw("error login in:", err);
    session = my_session;
}
```

Once logged you can fetch for files and folders using the session:

```
session.getUnitsAndFolders( function(units) {
    //units contain info about every unit and which folders it has
});

session.GetFiles( unit_id, folder, function( files ) {
    //info about the files in that folder
});
```

Check the `LFS.Session` class for more info about all the actions you can perform (create folders, units, give privileges, upload files, etc).

Also check the demo in the src folder to see an usage of the system.

Feedback

You can write any feedback to javi.agenjo@gmail.com