

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| **Document title** | **Tsugi App** |
| **Document reference number** | **UEL -** |
| **Project name** | **UNISA e-Content 20140363** |
| **Unisa Department** | **Department of Economics** |

Table of Contents

1 Description of the component 5

2 How to Install The component 5

3 how to use the component 10

Document sign-off

This document has been seen and accepted by the following people:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Date | Signature |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Compiler | Purpose | Change Summary |
| <yyyymmdd> | Name Surname | Document Created,  Internal Review,  External Review,  etc | Initial release,  Updates based on internal review,  Updates to section xxx,  etc |
|  |  |  |  |
|  |  |  |  |

# Description of the component

TSUGI

* What is Tsugi
* Installation and configuration

## Description of tsugi

Tsugi is a multi-tennant scalable LTI library and tool hosting environment. It is intended to make it more tractable to implement the Application Store that we will need for the [Next Generation Digital Learning Environment](http://www.ngdle.org/).

This repository is the Tsugi Administration, Management, and Developer Console. This code also implements an LTI 2.0 Provider, CASA App Store, and IMS ContentItem App store. Any Tsugi tool (including those written in Java and NodeJS) need to have this software installed and configured as pre-requisite. Perhaps someday we will build Java and/or NodeJS versions of the console - but for now we just use the PHP console for applications written in any language.

While earlier versions of this repository included a set of modules, examples, and even exercises, as we move towards a 1.0 release of Tsugi, these elements are now moved to separate repositories (see below).

# How to install it

## Pre-Requisites

* [Install GIT](https://github.com/EonConsulting/Tsugi-As-a-Stand-Alone-App/blob/master/docs/GITHUB.md) so that it works at the command prompt.
* Install a PHP/MySQL Environment like XAMPP / MAMP following the instructions at:

<http://www.wa4e.com/install.php>

* Check the code out from GitHub and put it in a directory where your web server can read it

git clone https://github.com/csev/tsugi.git

* Create a database and get authentication info for the database

CREATE DATABASE tsugi DEFAULT CHARACTER SET utf8;

GRANT ALL ON tsugi.\* TO 'ltiuser'@'localhost' IDENTIFIED BY 'ltipassword';

GRANT ALL ON tsugi.\* TO 'ltiuser'@'127.0.0.1' IDENTIFIED BY 'ltipassword';

* Copy the file config-dist.php to config.php and edit the file to put in the appropriate values. Make sure to change all the secrets. If you are just getting started turn on DEVELOPER mode so you can launch the tools easily. Each of the fields is documented in the config-dist.php file - here is some additional documentation on the configuration values:
* <http://do1.dr-chuck.com/tsugi/phpdoc/classes/Tsugi.Config.ConfigInfo.html>
* Go to the main page, and click on "Admin" to make all the database tables - you will need the Admin password you just put into config.php If all goes well, lots of tables should be created. You can run upgrade.php more than once - it will automatically detect that it has been run.

## Adding Some Tools

If you are just exploring Tsugi, or doing a developer bootcamp, you can add some tools from some of the other repositories:

* [A Few Useful Tsugi Tools](https://github.com/csev/tsugi-php-mod) This is a set of simple useful tools like a Map tool, Simple Gradebook, Peer Grader, PHP Autograder, and others. These were the original tools that were part of the Tsugi checkout. To restore these tools do the following

cd tsugi

git clone https://github.com/csev/tsugi-php-mod mod

and make sure to rerun the Admin / Database Upgrade step to create the tables for for the new modules.

* [Tsugi Module Sample Code](https://github.com/csev/tsugi-php-samples) - These are relatively short bits of code that you can look at as you write your own Tsugi Module.
* [Tsugi Developer Exercises](https://github.com/csev/tsugi-php-exercises) - This is a set of exercises of increasing difficulty suitable for a class or workshop. Working solutions are provided online. Source code for working solutions is only available to inctructors that contact Dr. Chuck.
* [Sample Tsugi Module](https://github.com/csev/tsugi-php-module) - Copy this if you want to start a fresh Tsugi Module from scratch. If you are building a new tool from scratch, you should build it as a "Tusig Module" following all of the Tsugi style guidance, using the Tsugi browser environment, and making full use of the Tsugi framework. This repository contains a basic "Tsugi Module" you can use as a starting point.
* [Sample Tsugi-Enabled Application](https://github.com/csev/tsugi-php-standalone) - You can also use Tsugi as a library and add it to a few places in an existing application. This repository contains sample code showing how to use Tsugi as a library in an existing application.

Each of these repositories contain instructions on how to install, configure, and hook each of these applications into your Tsugi instance. Once you install a new module or modules, you will need to re-run the Admin / Database Upgrade process to create the new tables required by the new applications.

Someday we will build a UI that can install these modules - but for now you must check them out and link them into Tsugi to function.

We have a short document on how to check out [all of the above tools](https://github.com/EonConsulting/Tsugi-As-a-Stand-Alone-App/blob/master/docs/CHECKOUT_ALL.md) and set up the configuration for them

## Tsugi implemented with LTI 2.0

2.3.1 Step 1

Go to <https://github.com/tsugiproject/tsugi> and clone the whole repo

2.3.2 Step 2

Copy the whole Tsugi folder in the root of your local server.

2.3.3 Step 3

Go to the browser and launch Tsugi , assuming you have already configured the confi.php

$CFG->pdo = 'mysql:host=127.0.0.1;dbname=tsugi';

// $CFG->pdo = 'mysql:host=127.0.0.1;port=8889;dbname=tsugi'; // MAMP

$CFG->dbuser = 'root';

$CFG->dbpass = '';

if running on mac OS, then change the following line

$CFG->mailsecret = 'warning:please-change-mailsecret-92ds29'; to $CFG->mailsecret = '';

Tsugi is already pre-configured with the LTI 2.0

2.3.4 Step 4

Login and upgrade the tables.

2.3.5 Step 5

LTI 2.0 has main features the registration services (URL), the proxies, the TCP/IP After some few hours exchanging with Mr John Tibbett we found out Tsugi has already been implemented with the LTI 2.0 and according to him Dr Chuck has alread amended the LTI 2.1 might be already available and implemented.

Tsugi has already the LTI 2.0 version inplemented

Open the file lti2.php, and find the line 64 and 74

// We have a person authorized to use LTI 2.0 on this server

$\_POST = $\_SESSION['lti2post'];

$lti\_message\_type = $\_POST["lti\_message\_type"];

$re\_register = $lti\_message\_type == "ToolProxyReregistrationRequest";

Right here the values of the URL are return according to what has been parsed

$return\_url\_status = false;

$return\_url\_tool\_guid = false;

$return\_url\_lti\_msg = false;

$return\_url\_lti\_errormsg = false;

The registration of the user logged must have its own key The key has to be unique and must not belong to other user And then we double check the registration of the scenarion in a transaction later on.

$tool\_proxy\_guid = false;

$tool\_proxy\_guid\_from\_consumer = isset($\_POST['tool\_proxy\_guid']);

if ( $re\_register ) {

$oauth\_consumer\_key = $\_POST['oauth\_consumer\_key'];

$reg\_key = $oauth\_consumer\_key;

$tool\_proxy\_guid = $oauth\_consumer\_key;

$tool\_proxy\_guid\_from\_consumer = true;

$key\_sha256 = lti\_sha256($oauth\_consumer\_key);

echo("key\_sha256=".$key\_sha256."<br>");

$oldproxy = $PDOX->rowDie(

"SELECT secret

FROM {$CFG->dbprefix}lti\_key

WHERE user\_id = :UID AND key\_sha256 = :SHA LIMIT 1",

array(":SHA" => $key\_sha256,

":UID" => $\_SESSION['id'])

);

$reg\_password = $oldproxy['secret'];

if ( strlen($reg\_password) < 1 ) {

lmsDie("Registration key $reg\_key cannot be re-registered.");

}

} else if ( $lti\_message\_type == "ToolProxyRegistrationRequest" ) {

$reg\_key = $\_POST['reg\_key'];

$tool\_proxy\_guid = isset($\_POST['tool\_proxy\_guid']) ? $\_POST['tool\_proxy\_guid'] : $reg\_key;

$oauth\_consumer\_key = $tool\_proxy\_guid;

$key\_sha256 = lti\_sha256($tool\_proxy\_guid);

echo("key\_sha256=".$key\_sha256."<br>");

$oldproxy = $PDOX->rowDie(

"SELECT user\_id

FROM {$CFG->dbprefix}lti\_key

WHERE key\_sha256 = :SHA LIMIT 1",

array(":SHA" => $key\_sha256)

);

if ( is\_array($oldproxy) && $oldproxy['user\_id'] != $\_SESSION['id'] ) {

lmsDie("Registration key $reg\_key cannot be registered.");

}

$reg\_password = $\_POST['reg\_password'];

} else {

echo("</pre>");

lmsDie("lti\_message\_type not supported ".$lti\_message\_type);

}

On the line 255 we found the registration URL, afterwards we dump the services

echo("<pre>\n");

$tc\_services = $tc\_profile->service\_offered;

echo("Found ".count($tc\_services)." services profile..\n");

if ( count($tc\_services) < 1 ) lmsDie("At a minimum, we need the service to register ourself - doh!\n");

If the URL registered is not found return the message otherwise return true by saying an application is found

if ( $register\_url == false ) lmsDie("Must have an application/vnd.ims.lti.v2.toolproxy+json service available in order to do tool\_registration.");

// unset($\_SESSION['result\_url']);

// if ( $result\_url !== false ) $\_SESSION['result\_url'] = $result\_url;

echo("\nFound an application/vnd.ims.lti.v2.toolproxy+json service - nice for us...\n");

Right here wee are ready to send the registration . There is a time to set up the key for the server side

$key\_sha256 = lti\_sha256($oauth\_consumer\_key);

echo("key\_sha256=".$key\_sha256."<br>");

echo("</pre>\n");

// Get the ack value

$ack = false;

if ( $re\_register ) {

$ack = bin2hex(openssl\_random\_pseudo\_bytes(10));

}

// Lets register!

$OUTPUT->togglePre("Registration Request",htmlent\_utf8($body));

$more\_headers = array();

if ( $ack !== false ) {

$more\_headers[] = 'VND-IMS-CONFIRM-URL: '.$CFG->wwwroot.

'/lti/tp\_commit.php?commit='.urlencode($ack);

}

$response = LTI::sendOAuthBody("POST", $register\_url, $reg\_key, $reg\_password, "application/vnd.ims.lti.v2.toolproxy+json", $body, $more\_headers, $hmac256);

$response\_code = Net::getLastHttpResponse();

global $LastOAuthBodyBaseString;

$OUTPUT->togglePre("Registration Request Headers",htmlent\_utf8(Net::getBodySentDebug()));

$OUTPUT->togglePre("Registration Request Base String",$LastOAuthBodyBaseString);

echo("<p>Http Response code = $response\_code</p>\n");

$OUTPUT->togglePre("Registration Response Headers",htmlent\_utf8(Net::getBodyReceivedDebug()));

This Jsinput component currently contains two pieces of related functionality:

It can author and deliver Jsinput GIFT-authored quizzes using LTI (index.php)

It can convert to QTI 1.2 (convert.php)

# How to use and Configure Tsugi

If you are just exploring Tsugi, or doing a developer bootcamp, you can add some tools from some of the other repositories:

[A Few Useful Tsugi Tools](https://github.com/csev/tsugi-php-mod) This is a set of simple useful tools like a Map tool, Simple Gradebook, Peer Grader, PHP Autograder, and others. These were the original tools that were part of the Tsugi checkout. To restore these tools do the following,

cd tsugi

git clone https://github.com/csev/tsugi-php-mod mod

and make sure to rerun the Admin / Database Upgrade step to create the tables for for the new modules.

[Tsugi Module Sample Code](https://github.com/csev/tsugi-php-samples) - These are relatively short bits of code that you can look at as you write your own Tsugi Module.

[Tsugi Developer Exercises](https://github.com/csev/tsugi-php-exercises) - This is a set of exercises of increasing difficulty suitable for a class or workshop. Working solutions are provided online. Source code for working solutions is only available to inctructors that contact Dr. Chuck.

[Sample Tsugi Module](https://github.com/csev/tsugi-php-module) - Copy this if you want to start a fresh Tsugi Module from scratch. If you are building a new tool from scratch, you should build it as a "Tusig Module" following all of the Tsugi style guidance, using the Tsugi browser environment, and making full use of the Tsugi framework. This repository contains a basic "Tsugi Module" you can use as a starting point.

[Sample Tsugi-Enabled Application](https://github.com/csev/tsugi-php-standalone) - You can also use Tsugi as a library and add it to a few places in an existing application. This repository contains sample code showing how to use Tsugi as a library in an existing application.

Each of these repositories contain instructions on how to install, configure, and hook each of these applications into your Tsugi instance. Once you install a new module or modules, you will need to re-run the Admin / Database Upgrade process to create the new tables required by the new applications.

Someday we will build a UI that can install these modules - but for now you must check them out and link them into Tsugi to function.

We have a short document on how to check out [all of the above tools](https://github.com/EonConsulting/Tsugi-As-a-Stand-Alone-App/blob/master/docs/CHECKOUT_ALL.md) and set up the configuration for them.

## Developer Documentation

You can view some of the developer documentation for the PHP version of Tsugi at:

* [Developer Documentation](https://github.com/EonConsulting/Tsugi-As-a-Stand-Alone-App/blob/master/docs/README.md)
* [PHP API Documentation](http://do1.dr-chuck.com/tsugi/phpdoc/)

## Other Repositories

The Tsugi Adminstration Console and Tsugi Modules / Applications depend on two other repositories:

[Tsugi PHP Library](https://github.com/csev/tsugi-php) - This is the code for the Tsugi run-time used by the Tsugi administration console and Tsugi PHP Modules and Applications.

[Tsugi Static Content](https://github.com/csev/tsugi-static) - This repository contains JavaScript, images, and CSS files shared across the various Tsugi implementations (PHP, Java, and NodeJS). The static content is available at <http://www.dr-chuck.net/tsugi-static/>- if you like you can check out your own copy  of this repo locally or for your production environment and point your Tsugi config.php at your own copy of the library.

## Other Languages

While the PHP Implementation of Tsugi is the most well developed, there are additional Tsugi implementations being developed:

[Tsugi Java](https://github.com/csev/tsugi-java-servlet) This is a reasonably complete implementation of the Tsugi run-time in Java. It shares low level IMS libraries with Sakai and is ready for production use.

[Tsugi NodeJS](https://github.com/csev/tsugi-node-sample) - This is early pre-emergent code.

## Tsugi Developer List

Please join the [Tsugi Developer List](https://groups.google.com/a/apereo.org/forum/" \l "!forum/tsugi-dev) so you can stay up to date with the progress regarding Tsugi.