

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

|  |  |
| --- | --- |
| **Document title** | **Graphs** |
| **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 5

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 component

# Graphs component is an easy to use component using the php stencil it adds the capability to draw interactive graphs or copy scripts from anywhere which can draw a graph to give the actual representation of the graph. It currently supports JSX Graphs.

# How to Install the SMS-Email Component

### Step 1.1

Install Laravel framework. The following command will install the latest version of Laravel with the project name LaravelDefault.

composer create-project laravel/laravel LaravelDefault

### Step 1.2

Download Zip from [Graphs repo](https://github.com/EonConsulting/Graphs).

## Step 2 - Configuration:

### Step 2.1

Create a folder called "Packages" inside of the root of the Laravel project.

### Step 2.2

Paste the folder from the Git repo (Step 1.2) inside the Packages folder.

### Step 2.3

Inside the root Laravel project, open the composer.json file. In that file, look for:

"autoload" : {

"psr-4": {

...

Inside of the psr-4 tag, load the following:

"EONConsulting\\Graphs\\": "Packages/Graphs/src"

### Step 2.4 - Use as a global function instead of by namespace (Optional)

Under the psr-4 but still inside of the autoload if the array files does not exist, create it like this:

"files": [

...

]

And enter into it:

"Packages/Graphs/src/Http/helpers.php"

## Step 3 - Registering the Service Provider

### Step 3.1

Open up config/app.php. In the providers array, enter the following:

EONConsulting\Graphs\GraphsServiceProvider::class,

### Step 3.2

In the same file, in the aliases array, enter:

'Graphs' => \EONConsulting\Graphs\src\Facades\Graphs::class,

## Step 4 - Tsugi

To be able to use the tsugi, the APP\_URL in your env file needs to be correct. So change it to the URL you are using for your testing environment.

## Step 5 - The Finale

In the command line / terminal, enter the following command in the root of the Laravel project:

composer dump-autoload -o

## Migrations

We have included a dummy\_table in the database for testing purposes. To have access to the migrations, you will need to perform the following commands:

php artisan vendor:publish

php artisan migrate

# How to use the Graphs component

## How to draw the graph

The Go to <http://localhost:8000/lecturer>

In the text area write the code to draw a graph. the code should be in the following format.

var b1 = JXG.JSXGraph.initBoard('jxgbox', {boundingbox: [-4, 2, 6, -4]});

// inside here write the implemantation of the function which will draw the graph.

If you dont follow the fomart the graph will not be previewed. Below is how you can draw a polygon using the text area.

// you can play arround with the bounding box coordinates which are in the format [x1,y1,x2,y2]

var b1 = JXG.JSXGraph.initBoard('jxgbox', {boundingbox: [-4, 2, 6, -4]});

var p1 = b1.create('point',[0,0], {name:'X',size:4});

var p2 = b1.create('point',[2,-1], {name:'B',size:4});

var p3 = b1.create('point',[-2,-3], {name:'C',size:4});

var p4 = b1.create('point',[-1,-1], {name:'D',size:4});

var p5 = b1.create('point',[3,1], {name:'E',size:4});

// Note that the "style" option is optional as is the parameter "name". Next we create a polygon through these five points "A" to "E".

var poly = b1.create('polygon',["X","B","C","D","E"]);

// Of course we can also use the JavaScript objects p1 to p5: to test it comment the above var poly statement and uncoment the below code and see the same result.

// var poly = b1.createElement('polygon',[p1,p2,p3,p4,p5]);

// the below code is critical as it is the one which updates your drawing board.

b1.update();

You can save the graph to the database using the save button which is below the text area and specifying the graph name.thats it the graph is saved.

The Examples page Shows different graphs you can drow copy and paste and see how it looks like