Land Portal manual

WESO Research Group

Contents

1 Installation manual

For a successful system installation the following elements are required:

- **Installation scripts**. To facilitate the installation task, and since this system has multiple internal and third-party components, the developers have created a series of scripts that

automate the system installation.

- *GNU/Linux distribution.* The installation scripts have been only tested under Ubuntu 14.04 LTS, so it is highly recommended to use this distribution for the system deployment.

The first step consists of logging into the system with a non-root account. The non-root account is important because the file paths in the scripts are relative to the user's home directory, the root home directory is */root* while the home directory for regular users is */home/USERNAME*.

Once you have logged into the system with a non-root account, the following step consists of copying the content of the scripts folder into the user's home directory. Such as in the image 1, the files *install.sh* and *settings.ph* and the folders *scripts* and [solr] must be located into the root of the user's home directory.
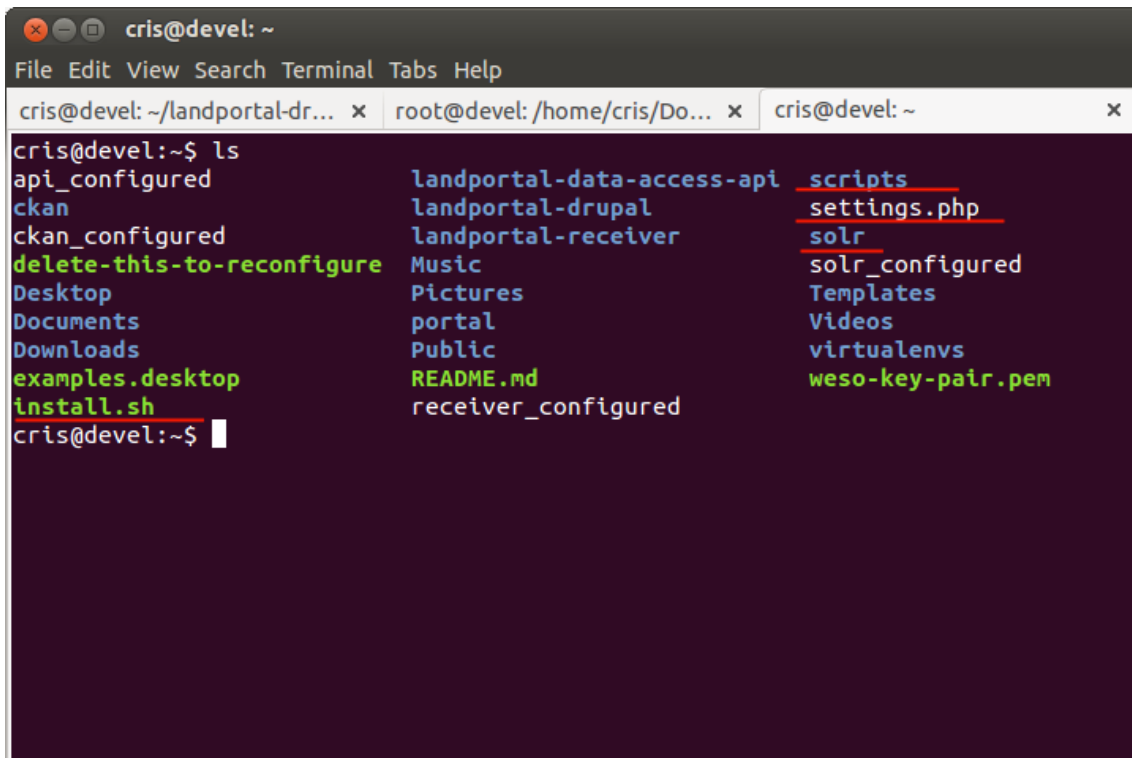

Figure 1: User's home folder with the script files (underlined in red)

After copying the installation scripts into the home directory, the system installation can be easily triggered with the command *sudo ./install.sh*, which means to run the script *install.sh*, which is located into the current directory, with superuser privileges. The superuser privileges are required to install some packages and configurations. The image 2 shows the command before starting the installation.

When you hit the *enter* key, the installation will begin. The installation process is completely automated and requires no user interaction, but since it is such a big system, the installation can last a long time.

After the installation ends, the system is completely functional, but it still needs some configuration. Please, take a look into the "Configuration manual".

Figure 2: Launch system installation command

2   Configuration manual

This configuration manual is also published online in the LandPortal's Drupal repository at Git-Hub[1].

Here are the instructions for configuring the new *LandPortal*. All of the following steps can be easily followed using the Drupal's administration interface. To use the administration interface you must log into the *LandPortal* using an account with administrator privileges.

Important notice: sometimes in this manual, there will appear some paths. Those paths are used to easily navigate for the Drupal's administration interface. Each path is relative to the current server host. This means that, if the current server host is *landportal.weso.es*, the path *admin/appearance* really means: *landportal.weso.es/admin/appearance*.

2.1   Enable the *LandPortal* theme

The new *LandPortal* appearance is provided by a custom theme created specially for the occasion. The theme receives the name of "*book*". To enable the "*book*" theme go to the tab *Appearance* in the top bar of the administration interface.

Once in the *Appearance* tab, scroll down to the bottom of the page, and in the section *disabled themes* click "*Enable and set default*" in the *Book theme for LandPortal*. Now, the "*book*" theme is used by default in all pages[2]

---

[1] https://github.com/weso/landportal-drupal
[2] For security reasons the administration interface will still use the default Drupal theme.

2.1.1 Configuring the *favicon*

The *favicon* is a little icon that shows in the browser's tabs and in the browser's bookmark section, representing the entire site. The new *LandPortal* has a nice *favicon* wich can be easily enabled:

1. Go to the path *admin/appearance/settings*

2. Go to the section *Shortcut icon settings* and uncheck the option *Use the default shortcut icon*

3. In the field *Path to custom icon* write **sites/all/themes/book/favicon.png**
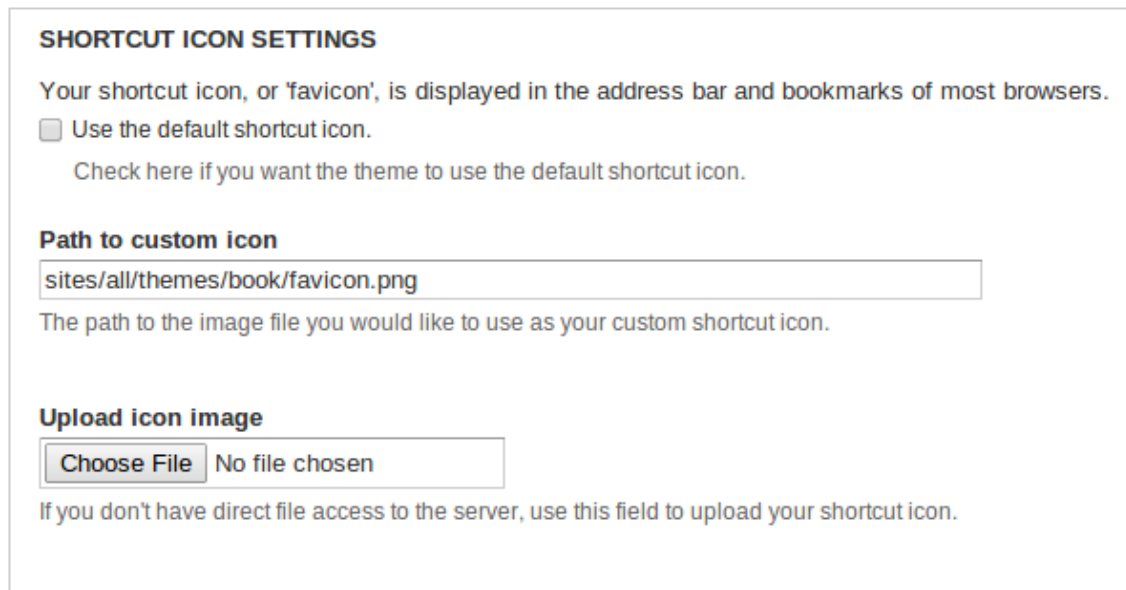
The image 3 shows how those options should look like.

Figure 3: *LandPortal*'s favicon configuration

2.1.2 Configuring *home* and *error* pages

The new *LandPortal* has a nice home and error pages. The home page is called *hub*, aid it has been designed to be the entry point to the new *LandPortal*. The error page shows a nice compass to entertain the users when an error happens. To enable those views take the following steps:

1. Go to the path *admin/config/system/site-information*

2. In the field *Default front page* write **home**

3. In the *ERROR PAGES* fields write **e404**

The image 4 shows how those fields should look like.

2.1.3 Configuring the *login* redirection

The default behaviour of Drupal consists of redirecting a user to it's profile page after he has logged into the system. As stated in the previous section, the *hub* page has been designed as the entry point for the new *LandPortal*. We want to override the Drupal's default behaviour to load our *hub* page after a user logs into the system.
Two steps are required to achieve this:

Figure 4: *LandPortal*'s home and error pages

1. Go to the path *admin/config/system/actions*. In the bottom page you will see the page creation form. Chose the option *redirect to URL* in the dropdown menu and click the button *Create*. A new page will load in which you can set the action label (we suggest something readable, for example: "*LandPortal redirect on login*"). In the field *url* write **home** (the *hub* path). The image 5 shows how those options should look.

2. Go to the path *admin/structure/trigger/user* and in the option *After an user has logged in* select the action created in the previous step. Then click the button *assign*.

Figure 5: User redirection configuration

2.2 Import the *taxonomy terms*

The *LandDebate* uses 5 different *taxonomies* or *vocabulary* to classify the different contents.

- Continents

- Countries

- Regions

- Debate status

- Topics

Each of those *taxonomies* is populated by *terms*. Unfortunately, those *terms* must be imported in a manual way (the import process only needs to be done once). The following steps explain how to import the *taxonomy terms*:

1. Go to the path *admin/structure/taxonomy*

2. Click the button *CSV IMPORT* in the right upper side

3. Choose *Translation* as the type of import. In the field *list of languages* write **und, es, fr**

4. Paste the taxonomy terms[3] into the text box called *Terms to import*

5. in the field *vocabulary choice* select the taxonomy **Continents**

6. Repeat the above steps with the **countries**[4], **topics**[5], **regions**[6] and **debate status**[7] taxonomies. Don't forget to select the corresponding destiny taxonomy in each case.

The images 7 and 6 illustrate some of the above steps.

Figure 6: Taxonomy import destiny vocabulary selection

2.3 Configure the content types

A new Drupal installation creates two content types by default, those content types are called *Article* and *Basic page* and will not be used into the new *LandPortal*, so they can be omitted or deleted without problem. The content types can be accessed in the path *admin/structure/types*.

2.3.1 Configure the *Blog posts*

The *Blog posts* content type can be configured in the path *admin/structure/types/manage/blog-post*

- Change the option *preview before submitting* to *disabled*

- In the section *comment settings* uncheck the options *Threading* and *Allow comment title*

---

[3]`https://github.com/weso/landportal-drupal/blob/develop/taxonomy_terms/continents.csv`
[4]`https://github.com/weso/landportal-drupal/blob/develop/taxonomy_terms/countries.csv`
[5]`https://github.com/weso/landportal-drupal/blob/develop/taxonomy_terms/topics.csv`
[6]`https://github.com/weso/landportal-drupal/blob/develop/taxonomy_terms/regions.csv`
[7]`https://github.com/weso/landportal-drupal/blob/develop/taxonomy_terms/debate_status.csv`

Figure 7: Taxonomy import type and languages configuration

- In the section *comment settings* set the option *Preview comment* to **Disabled**

2.3.2   Configure the *Debates*

The *Debates* content type can be configured in the path *admin/structure/types/manage/debate*

- Change the option *preview before submitting* to *disabled*

- In the section *comment settings* set the option *Default comment status for new content* to **Closed**

- In the section *comment settings* set the option *Preview comment* to **Disabled**

2.3.3   Configure the *Events*

The *Events* content type can be configured in the path *admin/structure/types/manage/event*

- Change the option *preview before submitting* to *disabled*

- In the section *comment settings* uncheck the options *Threading* and *Allow comment title*

- In the section *comment settings* set the option *Preview comment* to **Disabled**

- In the section *comment settings* set the option *Default comment status for new content* to **Hidden**

2.3.4   Configure the *News*

The *News* content type can be configured in the path *admin/structure/types/manage/news*. The configuration for this content type is the same as the configuration for the *Events*.

2.3.5   Configure the *Organizations*

The *Organizations* content type can be configured in the path *admin/structure/types/manage/organization*. The configuration for this content type is the same as the configuration for the *Events*.

2.4   Configure the search

2.4.1   Connect to the *Apache Solr* service

The new *LandPortal* uses *Apache Solr* to provide a high quality search service. The following steps are required to configure *Apache Solr*:

- Go to the *Configuration* tab in the top bar of the administration interface

- Go to the option *Apache Solr search* in the administration panel

- Go to the tab *Settings* and click the link named *edit*

- In the field *Solr server URL* write **http://localhost:8983/solr/drupal** and click the button *Save*

The image 8 shows how the *Apache Solr* search server should look after its configuration. The green colour means that Drupal has successfully contacted Solr.

Figure 8: Solr connection configuration result
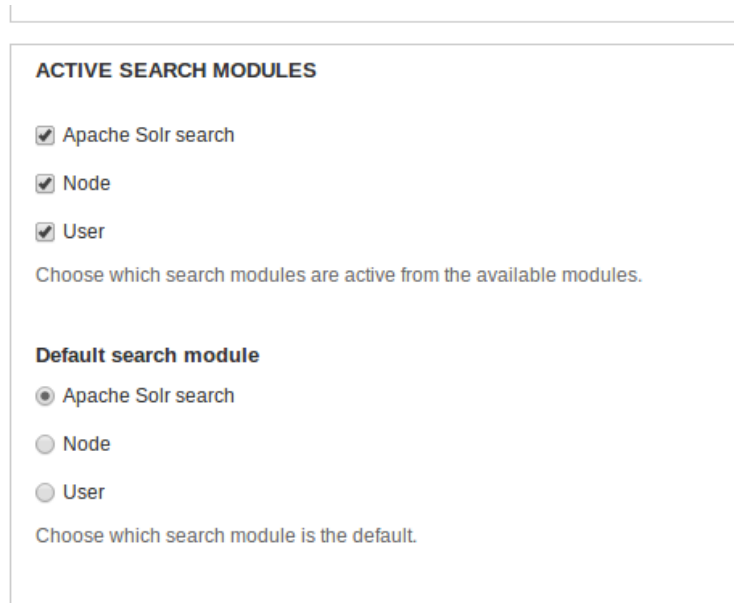
2.4.2   Set *Apache Solr* as the default search provider

Drupal supports multiple search providers. To use *Apache Solr* as the default search provide in the new *LandPortal* take the following steps:

1. Go to the path *admin/config/search/settings*

2. In the section *Default search module* choose the option *Apache Solr search*

The image 9 shows how those options should look like.
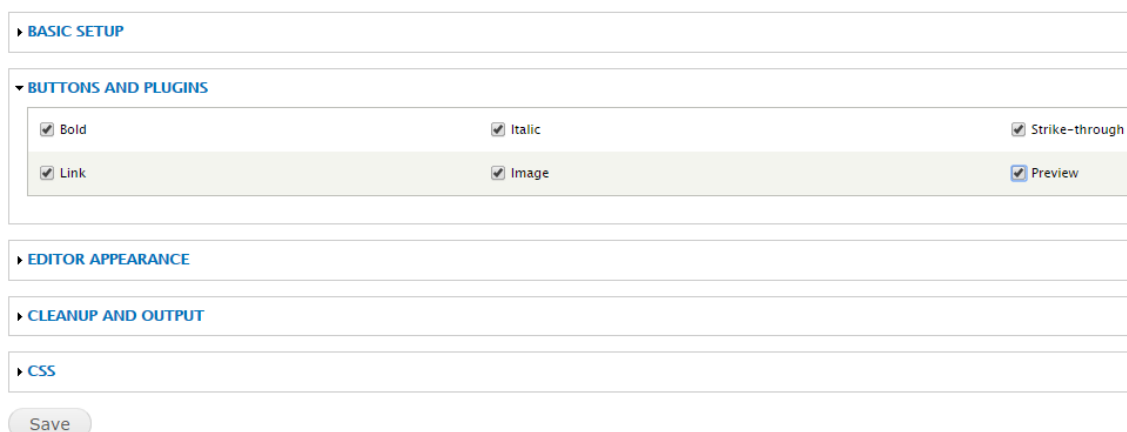
Figure 9: Solr as the default search server configuration

2.5 Configure the *WYSIWYG* editor

The *WYSIWYG* module allows Drupal to show a nice text editor component in which the users can easily format the text and insert images. To enable this module take the following steps:

1. Go to the path *admin/config/content/wysiwyg*

2. For each profile choose the editor *markItUp 1.1.14*

3. After selecting the editor you can change its options and choose which buttons to show. We suggest enabling all the buttons for the best user experience.

The image 10 shows how to enable those buttons.

Figure 10: WYSIWYG text editor configuration

3   User manual

This is the user manual for the new *LandPortal*, it has been structured as a set of questions and answers.

3.1   How do I start a new debate?

To start a new debate, you must load the debates view in the path */debate/debates*. If you are logged into the new *LandPortal* you will see a blue button to create a new debate in the right side.

After clicking that button the form to create a new debate will load. You must fill the form (the fields marked with an asterisk are required and can not be left empty) and click the *save* button. After clicking the *save* button (and if the form was correctly filled) the system will create the new debate.

By default all new debates are created with a "*coming soon*" status, which means that the comments are closed until the debate's period starts. When an administrator changes the debate status to "*open*" and opens the comments, the users will be able to post new comments and participate into the debate.

3.2   How do I close a debate?

This action is restricted to users with administration privileges.

To close an open debate you must load the detailed view of the debate that you want to close. In the bottom left you will see a blue button to edit the debate.

After clicking that button, the system will load a form (similar to the debate creation form) in which you can choose the correct settings for the debate. In the form, change the status of the debate to "*closed*" and close the comments. Click the *save* button to save the changes.

After a debate has been closed, the users can see the existing comments, but can not create new ones.

3.3   How can I register into the portal?

To create a new account into the new *LandPortal* you must load the home view and click the "*sign in*" button in the upper right corner. Once in the *sign in* view you can choose between a tab to log into the system with an existing account, or to create a new account.

Choose the tab to create a new account and you will see the register form. To create your account fill the form (the required fields are marked with an asterisk and can not be left empty) and click the button "*register*".

WARNING Your new account will not be usable right after the registration, For security reasons an administrator has to activate your account before you can log into the system.

3.4   How can I moderate new registrations?

This action is restricted to users with administration privileges.

By default, the new user registrations are in a "*disabled*" state, which means that the user can not log into the system. To enable the new user accounts, the admin must load the "*people*" tab

in the administration view. The *people* view shows a list of all user accounts (disabled or not) existing into the system. To activate the user accounts, the administrator has to select the users and click the *update* button.

3.5   How can I moderate articles?

This action is restricted to users with administration privileges.

The administrator can edit or delete any *article*[8]. To delete a certain *article* you must load its detailed view and click the button *edit* or *delete*.

The administrator can also edit or delete any user comments. To do this, you must click the button *edit* or *delete* that appears under every comment.

4   Tutorial: Creating a custom view

As Larry Garfield explains in [Gar06] Drupal uses the architectural pattern PAC[9] for organising its structure and functionality. The PAC pattern is less known and used than the architectural pattern MVC[10]. The new Land Portal tries to transform the PAC pattern to work in a way more similar to the MVC pattern.

In this tutorial we will explain how to create a custom view and include it in the new Land Portal. Those three steps are required and will be explained in detail.

1. Create the entry in the routes configuration file

2. Create the model that returns the required data

3. Create the view to show the data returned by the model

4.1   Creating the route

The first step to create a custom view is to create the route under the users will access the view. The routes are declared in the *routes.json* file. This file is stored in the path *DRUPAL_HOME/sites/all/modules/custom/mo*

The image 11 shows some entries of the *routes.json* file. Each entry has the following fields:

- **Path** The path specifies the URL of the view. The path is always relative to the main Drupal URL. To specify a *wildcard* use a % symbol. For example, in the image 11 the path *book/regions/%* will match the URLs *book/regions/1*, *book/regions/150*, etc

- **Name** The name is used to retrieve the corresponding model and view. The name *regions* will look for the model *Regions* and the view *regions.mustache*.

- **Title** The title is printed in the browser tab in which the user loads the template.

- **Navigation** The navigation sets the tab that will be selected in the header bar when te user loads the template.

- **Params** The params specify which parameters will be taken from the URL and passed to the model. The parameter corresponds to the position of the *wildcard* in the *path*, the position

---

[8]In this context, *articles* refers to any content in the social section of the LandPortal, such as news, events, debates, etc.

[9]Presentation Abstraction Control - `http://en.wikipedia.org/wiki/Presentation-abstraction-control`

[10]Model View Controller - `http://martinfowler.com/eaaDev/uiArchs.html`

starts in 0. For example, in the image 11 the path *book/regions/%* has the *wildcard* in the position 2, being the position 0 *book* and the position 1 *regions*. If a path does not receive any arguments leave the parameters empty.

- **Redirect** Apart from the previous fields, an entry in the routes file may have a *redirect* field. This field contains a path that will be loaded automatically. For example, in the image 11 the path *book* will redirect to the path *book/regions*. When the *redirect* field is present, the fields *navigation* and *params* can be omitted.

```
37              },
38              {
39                      "path": "book/regions",
40                      "name": "regions",
41                      "title": "Region list",
42                      "navigation": "regions",
43                      "params": []
44              },
45              {
46                      "path": "book",
47                      "name": "book",
48                      "title": "Land Book",
49                      "redirect": "book/regions"
50              },
51              {
52                      "path": "book/regions/%",
53                      "name": "regions",
54                      "title": "Region",
55                      "navigation": "regions",
56                      "params": [2]
57              }.
```

Figure 11: Snippet of the *routes.json* file content

The image 12 shows a custom created route for this tutorial. The route will be accessible in the path *helloworld* and will load the model *helloworld.php* and the template *helloworld.mustache*. In the following sections we will see how to create the model and the template.

```
113 ▼           {
114                     "path": "helloworld",
115                     "name": "helloworld",
116                     "title": "Hello World!",
117                     "navigation": "",
118                     "params": []
119             }
```

Figure 12: Example of new entry in the *routes.json* file

4.2   Creating the model

The second step to create a custom view is to create a model. The models return the data that is shown to the user by the template. The models are created in the directory *DRUPAL_HOME/sites/all/modules/custom* The model will be automatically loaded by its name.

In the image 12 we created a new path to make a custom template. The path had the name *helloworld*, so our model will have the name *helloworld.php* in order to be automatically loaded by Drupal.

The image 13 shows the content of our new model. Every model must have two basic characteristics in order to be correctly loaded by Drupal:

- The class name of the model must be the same as the file name, but with the first letter upercase. Our model file was called *helloworld.php* so the model's class must be called *Helloworld*.

- Every model must have a method *get* that returns an array with the contents to populate the template. In this case the model returns a single key called *name*, which will be rendered in the template.

```php
 1  <?php
 2
 3  class Helloworld {
 4
 5      public function get() {
 6          return array(
 7              "name" => "Land Portal"
 8          );
 9      }
10  }
```

Figure 13: Example of the model class in the *helloworld.php* file

4.3 Creating the template

The final step to create a custom view is to create the template that will be rendered to the user. The templates are created in the directory *DRUPAL_HOME/sites/all/themes/book/views/*. The template will be automatically load by its name.

In the image 12 we created a new path to make a custom template. The path had the name *helloworld*, so our template will have the name *helloworld.mustache* in order to be automatically loaded by Drupal.

The templates are created using *mustache*[11]. The arguments between the mustaches ({{ and }}) will be automatically replaced by its value before the template is rendered to the user.

The image 14 shows the contents of our custom template in the file *helloworld.mustache*. The model that we created in the image 13 returned a key named *name* that we will use in the template.

The image 15 shows the result of the view that we have created in this tutorial. As you can see, the path corresponds to the one specified in the image 12, and the mustaches in the image 14 have been substituted by the data returned from the model (image 13).

```html
 1  <h1>
 2      Hello {{name}}!
 3  </h1>
```

Figure 14: Example of the template in the *helloworld.mustache* file

---

[11]It is not the object of this tutorial to explain *mustache* and it syntaxis. For more details about how *mustache* works, take a look at the official documentation in `http://mustache.github.io/`
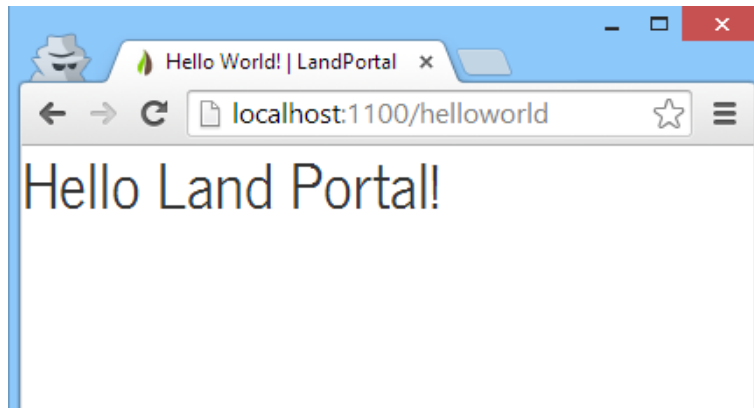
Figure 15: Result of the custom view

References

[Gar06]   Larry Garfield. 'MVC vs. PAC'. In: (2006). URL: http://www.garfieldtech.com/blog/mvc-vs-pac.