# Setup and installation of the Accenture Recruiter website

The links in this doc assumes your localhost run on port 8888. If not, you need to change your links from localhost:8888 to localhost:80 or localhost:81, depending on your setup.

**Wordpress**

The website will run on <https://recruitment.innovationtrampoline.com>. I will do the deployments, as I can’t create more FTP accounts.

<https://recruitment.innovationtrampoline.com/wp-admin> for admin screens.

Username is xm (**WnkNrxrB$9MfEU93JI#JocPZ**) and leslie (**bs$O5%tzDOTb!eUADo69sHiq**). Password in **bold.**

Then get the theme files from Github. The files should go into your Wordpress folder under **/wp-content/themes/**. The folder in GitHub is <https://github.com/rohanstrydom/recruitment>. Under your Wordpress installation you should now have **/wp-content/themes/recruitement**. I have also created a folder called **recruitment-fe** where you should please upload your front-end code. Where you put this is irrelevant for Wordpress.

You need to install three plugins.

1. Advanced Custom Fields: <https://wordpress.org/plugins/advanced-custom-fields/> . You can then import the**/recruitment/wordpress-setup/advanced-custom-fields.json**file (found in the theme download) at<http://localhost:8888/wp-admin/edit.php?post_type=acf-field-group&page=acf-tools>.
2. Custom Post Type UI: <https://wordpress.org/plugins/custom-post-type-ui/>. Then go to <http://localhost:8888/wp-admin/admin.php?page=cptui_tools> on your local Wordpress instance. You then then import the contents of the**/recruitment/wordpress-setup/custom-post-type-ui.txt file**.
3. Protect Uploads: <https://wordpress.org/plugins/protect-uploads/>.

You will see an message that you can upgrade Advanced Custom Fields to version 5. Please do.

**Wordpress API’s**

These files are all defined in a Wordpress file /wp-content/themes/recruitment/functions.php. I will update the **/wp-content/themes/recruitment/wordpress-setup/api-definitions.txt** file with all the paramaters and return values. Currently the following are defined…

add\_action('wp\_ajax\_candidate\_save\_phone\_number', 'candidate\_save\_phone\_number');// for users who are logged in \*\*\* done \*\*\*

add\_action('wp\_ajax\_nopriv\_candidate\_save\_phone\_number', 'candidate\_save\_phone\_number');//for users that are not logged in. \*\*\* done \*\*\*

add\_action('wp\_ajax\_candidate\_save\_otp', 'candidate\_generate\_save\_otp');// for users who are logged in \*\*\* done \*\*\*

add\_action('wp\_ajax\_nopriv\_candidate\_save\_otp', 'candidate\_generate\_save\_otp');//for users that are not logged in. \*\*\* done \*\*\*

add\_action('wp\_ajax\_candidate\_check\_otp\_otp', 'candidate\_check\_otp');// for users who are logged in \*\*\* done \*\*\*

add\_action('wp\_ajax\_nopriv\_candidate\_check\_otp', 'candidate\_check\_otp');//for users that are not logged in. \*\*\* done \*\*\*

add\_action('wp\_ajax\_candidate\_save\_detail', 'candidate\_save\_detail');// for users who are logged in

add\_action('wp\_ajax\_nopriv\_candidate\_save\_detail', 'candidate\_save\_detail');//for users that are not logged in.

add\_action('wp\_ajax\_candidate\_save\_response\_one', 'candidate\_save\_response\_one');// for users who are logged in

add\_action('wp\_ajax\_nopriv\_candidate\_save\_response\_one', 'candidate\_save\_response\_one');//for users that are not logged in.

add\_action('wp\_ajax\_candidate\_save\_response\_two', 'candidate\_save\_response\_two');// for users who are logged in

add\_action('wp\_ajax\_nopriv\_candidate\_save\_response\_two', 'candidate\_save\_response\_two');//for users that are not logged in.

add\_action('wp\_ajax\_candidate\_save\_response\_three', 'candidate\_save\_response\_three');// for users who are logged in

add\_action('wp\_ajax\_nopriv\_candidate\_save\_response\_three', 'candidate\_save\_response\_three');//for users that are not logged in.

With your REST client you can access the api’s on your local machine at <http://localhost:8888/wp-admin/admin-ajax.php> and on the server at <http://recruitment.innovationtrampoline.com/wp-admin/admin-ajax.php>. Below is an example to check the one time password (OTP).

cid:image001.png@01D3D7D9.288D4BA0

Testing API’s

There is another way to test the API’s, and this is from a Wordpress index.php page. If you go to <http://localhost:8888> (my localhost MAMP setup), you will actually run the /wp-content/themes/recruitment/index.php file. Where are the API calls will expect paramaters to be posted, when you run the index.php, you need to change the function file to NOT expect $\_POST[????] values, but rather set variable values. For instance…

/wp-content/themes/recruitment/functions/candidate\_check\_otp.php has this code to make it work on a api call.

$phone\_number = $\_POST['phone\_number'];

        $position\_code = $\_POST['position\_code'];

        $otp = $\_POST['otp'];

        //$phone\_number = '0823749048';

        //$position\_code = '0DIGB0E020180413';

        //$otp = '123457';

In order to make it work through the index.php page, uncomment the Green lines, and comment out the blue lines.

        //$phone\_number = $\_POST['phone\_number'];

        //$position\_code = $\_POST['position\_code'];

        //$otp = $\_POST['otp'];

        $phone\_number = '0823749048';

        $position\_code = '0DIGB0E020180413';

        $otp = '123457';

In the /wp-content/themes/recruitment/indexp.hp file, you need to uncomment the function you need to run. In our example…

// candidate\_save\_phone\_number(); // working

// candidate\_generate\_save\_otp(); // working

candidate\_check\_otp();  // working