Final Project Report

Team Zoundi

Ali Arshad, Arnold Zoundi, David Szabo, Ryan Hillier

April 3rd, 2017

Project Summary

The Romeo Dallaire Child Soldiers Initiative (RDCSI) created the Six Stories project to help increase global awareness of children being used as soldiers. The Initiative hopes to completely end the practice through its awareness campaigns and charitable work. RDCSI requested the development of an interactive, storytelling website to aid in reaching this goal. This website will serve as an online educational tool for those who are interested in learning more about the Initiative's mission.

Our team was tasked with creating a Wordpress REST API that will be connected to a customized UI (built with angular.js). We were also asked to create dummy data to provide proof of concept that the WP REST API was functional in conjunction with the side scrolling assets developed by the RDCSI. We were able to successfully create the API that uses the WP content management system to connect endpoints and show content. This was completed by the midterm presentation. We then worked on creating a website to demo our implementation of the REST API, and made several changes in our API to better support the project.

We created three demos to show the functionality of the REST API: one that represents the parallax effect combined with the PSDs that the client gave us, one that shows how we will get information from the REST API, and one that can be seen as a polished, final proof of concept of what the final website could look like. Although we faced many challenges throughout the course such as illness, and loss of team members to an already small group, we were able to push through and complete all of the major user stories in time for the end of the course.

Description

Six Stories is an interactive, and user driven project with the goal of educating its users in a memorable and interesting way. The project hopes to raise awareness and help users better understand the effects of child soldiers on children, communities, and nations. Interactivity is one of the most important pieces of this project. Users choose one of six different characters who represent different stakeholders who are affected, such as a child soldier, a peacekeeper etc. Once a character has been chosen, the user begins a journey through that character's life, making decisions which directly impact the story. As the user scrolls the story advances, moving between discrete events, implemented as "posts". These posts can contain text, images, audio and video. Each character thread is meant to intertwine with the others, leading to events where a peacekeeper may meet a child soldier. Many events can only take place if certain decisions were made by the user previously. Users can swap characters at meeting points, and view how their actions have affected this character.

Our group's task was to create a WordPress REST API which could provide support to a front end developer who will be implementing their artistic vision for the Six Stories. This entailed building a flexible, efficient, and easy to use backend. Our main objectives were:

(I) Creating a flexible API which allows a developer to implement their vision without having to compromise due to technical limitations.

(II) The entire API runs efficiently and is maintainable and extendable should others need to make changes.

(III)The API is created using the technology prefered by the client.

Planned vs. Actual

Initially, we were tasked with developing a WordPress REST API that the RDCSI could send to a front end developer to create the interactive Six Stories website. To complete the REST API, we had the possibility of using existing plugins available through WordPress but decided to create our own plugin to handle the functionality of creating custom post types. The creation of custom routes was achieved with the use of a third party WordPress plugin. We were able to complete the REST API by the midterm stage of the course, and thus were able to expand the project's scope.

At this point, we were also tasked with developing a website that could act as a proof of concept that the REST API was working as designed. We were able to successfully complete the proof of concepts, showing how the REST API call functions work, and the integration of the PSDs with the existing parallax effect that was already developed. We also created a website that shows a possibility of what the final web page could look.

Original Scope / Scope Change

The team has made great progress on the Six Stories project since the start of the execution phase. The table below details the status of the master user story list in relation to the project plan as of April 3rd, our submission date for this project .

User Story Initial

User Story	Estimation(Hrs)	Points	Status
As a front-end developer, I want to be able to implement the Six Stories in an interactive way so that we can spread awareness on the issue of child soldiers worldwide.	16	5	Completed
As a primary stakeholder, I want to be able to create any narrative and have a flexible backend which supports it in order to save me time and money.	8	3	Completed
As a primary stakeholder, I want to be able to easily understand and use the API that is	8	3	Completed

developed in order to facilitate the creation of content for my web site.			
As a user I want to be able to have a fast and interactive experience in order to better facilitate understanding of the Romeo Dallaire Child Soldiers Initiative's impact.	5	1	Completed

Table 1.0 Initial User Story

The first user story was completed and is represented by using a dummy website that would preview all the features of the six stories website. The website includes concept art and dummy data and helps convey a generic idea how the six stories project could look once a front end developer has completed their work,

The second user story has been completed. The client will be able to create content in the backend by simply creating posts through WordPress. The client would like to store and display stories related to six characters. Our team created six custom post types which have the same features as a regular WordPress post types with extra custom fields. This interface design allows anyone to easily create and store content which will be accessible through the WordPress REST API.

The third user story has been completed. Our team met with the client during the execution phase to present the work completed up until that point. He provided us with positive feedback on the REST API interface design and the overall direction the project had moved.

The fourth user story is completed. Our team has completed the process of modifying the Custom Post Types to allow the intersection of different stories and multiple branches. A front end developer should have no trouble using this system.

All user stories were completed in time for the final presentation so there were no major changes in scope. After the midterm presentation the client was quite satisfied with the progress that the team had made and suggested some additional deliverables for the project that have been explained in the deliverables section of the project report.

Major Challenges

The team encountered many issues during the development of this project such as the authentication to the WordPress REST API, the implementation of custom routes in the six stories plugin. Many of these issues were related to a single challenge which is the lack of learning resources for the WordPress REST API .

The only major challenge the team encountered while working on this project was the lack of proper learning resources. The team considered different resources such as WordPress official REST API Handbook and online tutorials written by individuals and experts in the WordPress community.

WordPress provides an online handbook of their REST API to help developers make products which use the API features. However, it does not provide enough examples and details on the implementation of some of the API functions. In fact, the documentation seems to be more suitable to individuals with prior experience with WordPress. The team was able to make use of the information in the documentation to implement simple features such as custom post types.The implementation of custom routes as a standalone plugin required a deeper understanding of WordPress functionalities. As a result, the team used a third party plugin to implement them.

Online tutorials were good alternative to the WordPress REST API documentation.However, these still suffered from not providing enough details. In fact, most of these tutorials provided code snippets targeted to standard WordPress posts instead of custom post types. As a result, the team spent a lot of time trying to figure out how to adapt these snippets to the project. This task was not trivial and required the team to put in a huge amount of time and effort to complete the project's requirements such as creating custom routes for custom post types.

Deliverables

Polished website as a proof of concept.

The Website is a proof of concept for the project and shows how the website could look like once a front end developer has worked on it. The website includes dummy data with pictures of six different characters and background images provided by the client.

Custom post types plugin

The six stories plugin encapsulates the logic behind the implementation of custom post types for each character in the storyline. It is already installed and activated in the supplied WordPress development environment.

Product Documentation

The product documentation included in this report mainly guides the client and any developer through the creation of a post using the available custom post types. In addition, it provides insights in how a developer can query post information and make stories intersect during the timeline.

Updated custom parallax

The custom parallax page created with the help of modifying the previous groups work. The parallax includes the background images made from the psds provided by the client. The parallax gives a generic idea of the most important function of the website i.e using a character to scroll around a map on the website.

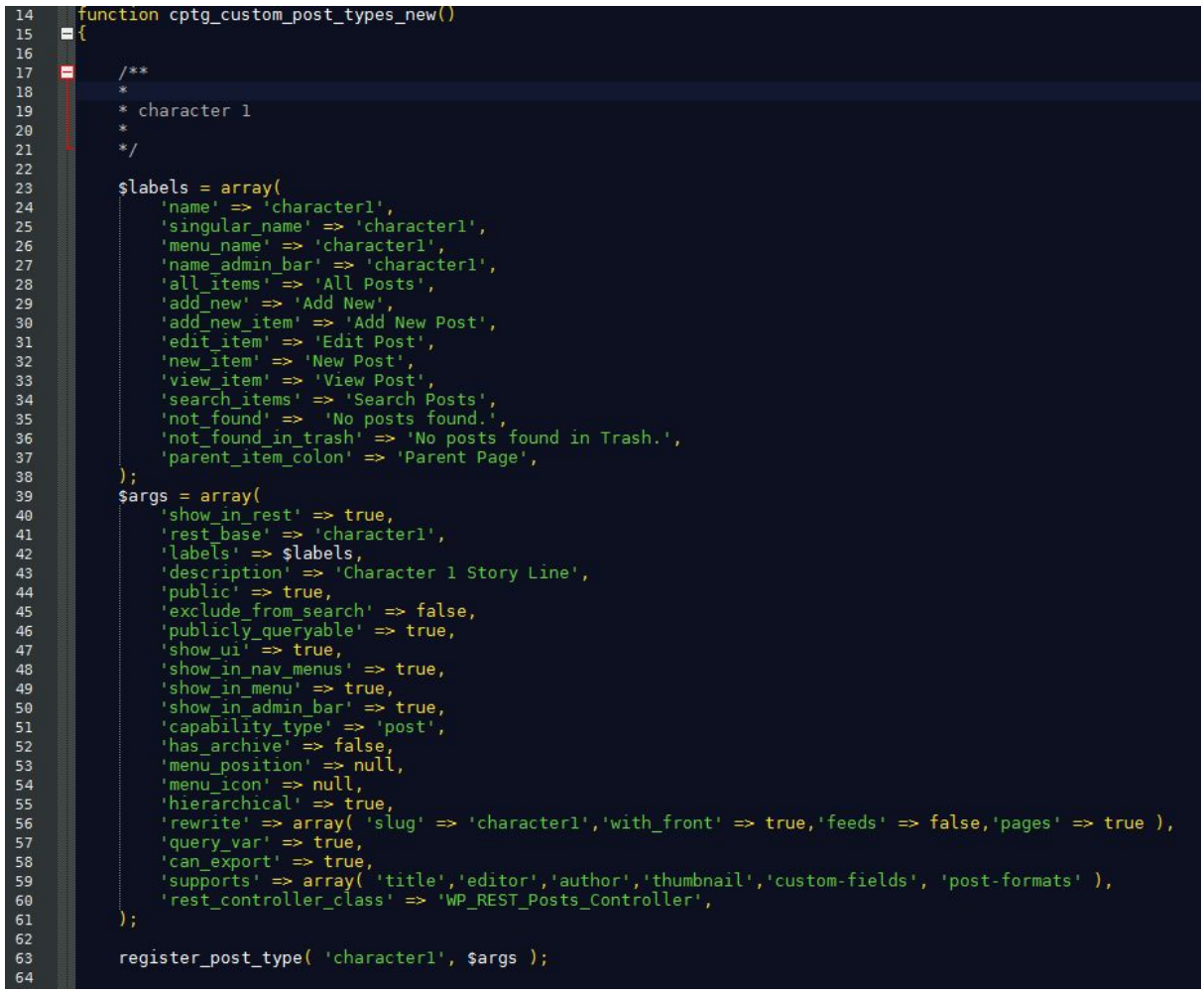
Product/System documentation

Custom Post Types

The design of the custom post types for this project revolves around the different characters in the storyline. This means each custom post type represents a specific character in the storyline. The custom post types have similar properties to the standard WordPress posts which means the client will be able to create a story for a specific character by simply creating a WordPress Post. Below is the breakdown of the six custom post types we implemented in this project:

- Custom Post Type 1: character1
- Custom Post Type 2: character2
- Custom Post Type 3: character3
- Custom Post Type 4: character4
- Custom Post Type 5: character5
- Custom Post Type 6: character6

The names used for these custom post types are not an actual representation of characters' names in the storyline. These are names used for development and testing purposes. However, the client or any web developer will easily be able to change these names to whatever is preferred. The plugin provided as part of the deliverables is properly structured to allow easy modifications of the different custom post types.



```
14  function cptg_custom_post_types_new()
15  {
16
17      /**
18      *
19      * character 1
20      *
21      */
22
23      $labels = array(
24          'name' => 'character1',
25          'singular_name' => 'character1',
26          'menu_name' => 'character1',
27          'name_admin_bar' => 'character1',
28          'all_items' => 'All Posts',
29          'add_new' => 'Add New',
30          'add_new_item' => 'Add New Post',
31          'edit_item' => 'Edit Post',
32          'new_item' => 'New Post',
33          'view_item' => 'View Post',
34          'search_items' => 'Search Posts',
35          'not_found' =>  'No posts found.',
36          'not_found_in_trash' => 'No posts found in Trash.',
37          'parent_item_colon' => 'Parent Page',
38      );
39      $args = array(
40          'show_in_rest' => true,
41          'rest_base' => 'character1',
42          'labels' => $labels,
43          'description' => 'Character 1 Story Line',
44          'public' => true,
45          'exclude_from_search' => false,
46          'publicly_queryable' => true,
47          'show_ui' => true,
48          'show_in_nav_menus' => true,
49          'show_in_menu' => true,
50          'show_in_admin_bar' => true,
51          'capability_type' => 'post',
52          'has_archive' => false,
53          'menu_position' => null,
54          'menu_icon' => null,
55          'hierarchical' => true,
56          'rewrite' => array( 'slug' => 'character1','with_front' => true,'feeds' => false,'pages' => true ),
57          'query_var' => true,
58          'can_export' => true,
59          'supports' => array( 'title','editor','author','thumbnail','custom-fields', 'post-formats' ),
60          'rest_controller_class' => 'WP_REST_Posts_Controller',
61      );
62
63      register_post_type( 'character1', $args );
64
```

Figure 1: character1 custom post type code snippet

Figure 1 represents the code snippet we wrote to create the custom post type for character one. The client or the web developer can easily replicate this snippet to create an additional custom post type for the project. However, they have to make sure to change the value on lines: {24, 25, 26, 27, 41, 43, 56, 63} to reflect the name of the new custom post type. For instance, the creation of a new custom post type called "Child" will require the values being replaced for the given lines with that String.

Custom Routes

The main objective of custom routes is to provide the client with customized paths to query specific information from the REST API. The team used the WordPress plugin “WP Rest Routes” to create the custom routes for the different custom post types. These routes make it possible to return custom fields information from specific posts. WordPress provides custom fields on post to allow users to add extra information as Key/Value pairs. The custom fields the client should consider using when creating a new post are the following: custom_id, format, incoming_connections, and outgoing_connections. The standard REST routes provided by WordPress do not include information entered in the custom field section. However, this was made possible the plugin “WP Rest Routes”.

In order to create a custom route using this plugin, a developer should:

1. Make sure the plugin is installed and activated
2. Go to the plugin page
3. Click on the “Add new” button to create a new route
4. Provide a name for your route. It will be part of the route’s complete URL
5. Build your query by selecting the post type you would like this plugin to be associated with. For instance, you can choose “character1”.
6. Build the content output by selecting the desired information to be returned by the route
7. Click on the “Publish” button to save the plugin.
8. Scroll to the bottom of the page to find out the route’s URL

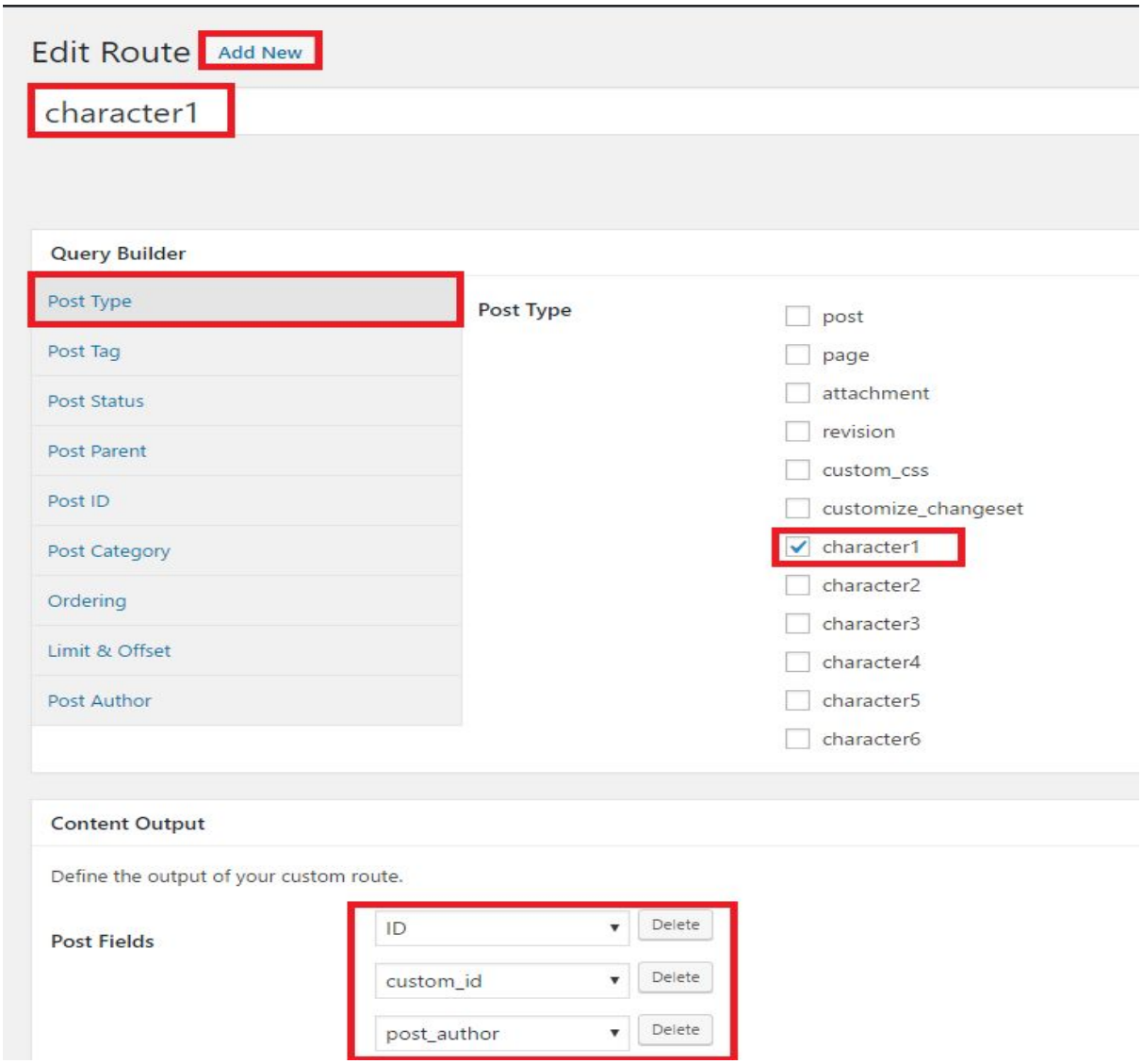


Figure 2: Custom Routes Creation

Figure 2 provides a representation of the steps mentioned above which will lead to the creation of a custom route for a custom post type such as “character1”.

The team created the following plugin for the six stories project. It is worth mentioning, a web developer can easily add additional routes as needed:

- Route http://dev-sixstories.pantheonsite.io/wp-json/rest-routes/v2/ch1 for character1
- Route http://dev-sixstories.pantheonsite.io/wp-json/rest-routes/v2/ch2 for character2
- Route http://dev-sixstories.pantheonsite.io/wp-json/rest-routes/v2/ch3 for character3
- Route http://dev-sixstories.pantheonsite.io/wp-json/rest-routes/v2/ch4 for character4

- Route http://dev-sixstories.pantheonsite.io/wp-json/rest-routes/v2/ch5 for character6
- Route http://dev-sixstories.pantheonsite.io/wp-json/rest-routes/v2/ch6 for character6
- Route http://dev-sixstories.pantheonsite.io/wp-json/rest-routes/v2/chall for all characters information

These routes will return the following information in a JSON Object:

- The ID of the post provided by WordPress
- The custom ID of the post entered by the client in the custom field section. This field gives more control over the way post are organized in WordPress ecosystem.
- The post author
- The post date
- The post title
- The post content
- The post format
- The incoming connections of a post provided by the client in the custom field section. This will mainly be used to facilitate stories crossover.
- The outgoing connections of a post provided by the client in the custom field section. This will mainly be used to facilitate stories crossover.

The modification of these custom routes cannot be easily achieved. In fact, the plugin does not seem to return modified or added information for a specific custom route. whenever the information in an existing route is modified to include information such as a new custom field, the URL of that route will always display the same information before the modification. In order to make sure these changes are committed, the WordPress admin (client or web developer) has to completely create a new route with a different name.

Intersections

Story intersection support is one of the major considerations we had when building the custom post types. Custom post types support incoming and outgoing connections to better model intersections. Front end developers can then implement a story structure where a particular post links to a post in another character's story allowing for intersections.This way, a player can move through all the stories at intersections, and experience what has occurred with a character since they last interacted with that character.

Signatures / Sign-off

By signing below you are agreeing that Team Zoundi has completed what was stated in the project's scope for the WInter 2017 semester and has handed the project back you, the client.

Signature _____ Date _____

Signature _____ Date _____