Observed Data App using the Tethys Platform

Matt Saguibo

A project paper submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

E. James Nelson, Chair
Daniel P. Ames
Norm Jones

Department of Civil and Environmental Engineering
Brigham Young University
April 2015

Copyright © 2015 Matt Saguibo
All Rights Reserved
ABSTRACT

Observed Data App using the Tethys Platform

Matt Saguibo
Department of Civil and Environmental Engineering, BYU
Master of Science

This paper demonstrates the creation of a web app that retrieves data from HydroServer stored on HIS-Central. The creation of the app fulfilled the two primary objectives, namely: 1. Demonstrate that someone with a general programming skill could create and publish an app with Tethys, and 2. Create an app that can retrieve, plot and print the Dominican Republic data from a Hydrosaver. The web app was developed with a program called Tethys Platform, which was developed at BYU by Nathan Swain as part of the NSF CI-WATER grant. This platform is used to develop and host water resources web applications otherwise known as “apps”. The reasoning behind Tethys was “to make the development of engaging, interactive web apps for water resources as easy as possible” (Swain, 2015). The Observed Data app can be found on the Tethys demo server page. Several apps have been developed on Tethys and it is expected to have a vast library of apps in the future. The Tethys Platform can be a gateway to creating a cloud-based water resource data infrastructure. This project hopes to serve as a template for how Tethys Apps can be developed.

Keywords: Tethys Platform, HydroServer, App
ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1135482. This project is based largely on material developed by Nathan Swain. His aid and guidance helped allow this project to continually progress and be completed. Additionally, the project was under the supervision of Dr. E. James Nelson whose open perspective facilitated work that was accomplished. Also, the shared data was made possible thanks to the people at INDHRI of the Dominican Republic. Lastly, my wife for the love and support she provided me throughout my education.
# TABLE OF CONTENTS

LIST OF FIGURES ........................................................................................................... vii

LIST OF TABLES ............................................................................................................... ix

1 Introduction ...................................................................................................................... 1

1.1 Purpose ...................................................................................................................... 2

1.2 Data Observations and Storage .............................................................................. 2

2 Tethys Platform .............................................................................................................. 5

2.1 Tethys Gizmos and Templates ............................................................................... 6

3 App Development .......................................................................................................... 9

3.1 Programming Background .................................................................................... 9

3.2 Install Tethys Platform ......................................................................................... 10

3.3 Perform the Tethys Tutorials ............................................................................... 11

3.4 Library of Gizmos ................................................................................................. 14

3.5 Create the New App Scaffold ............................................................................. 15

3.6 Begin New App Development ............................................................................. 15

3.7 GitHub Repository ............................................................................................... 18

3.8 Tethys App Demo Server .................................................................................... 18

4 Project Outcome ......................................................................................................... 19

4.1 Tethys Platform Outlook ..................................................................................... 19

4.2 Observed Data App Outlook .............................................................................. 20

REFERENCES .................................................................................................................. 21
LIST OF FIGURES

Figure 1-1: Key Components of CUASHI-HIS (*HydroServer*, 2015) ..............................................2
Figure 2-1: Tethys Software Platform (Swain, 2015)........................................................................5
Figure 2-2: Tethys Apps Library (Swain, 2015)...............................................................................6
Figure 2-3: User Interface Elements known as “Gizmos” (Swain, 2015).................................7
Figure 2-4: Canned GSSH A Tethys App (Swain, 2015)...............................................................8
Figure 3-1: app.py file portions used for the Observed Data App..................................................12
Figure 3-2: Home.html file used for the Observed Data App.......................................................13
Figure 3-3: Home page of the Observed Data App .................................................................13
Figure 3-4: Tethys Developer Tools (Swain, 2015)..................................................................14
Figure 3-5: Map of the Observed Data App ..................................................................16
Figure 3-6: Input Form of the Observed Data App ..................................................................17
Figure 3-7: Plot Page of the Observed Data App .................................................................18
LIST OF TABLES

Table 3-1: Programming Language Skill Levels ................................................................. 10
1 INTRODUCTION

The use of data is an every day occurrence for engineers. Data is used in a variety of manners from assumptions to calculations and models to evaluations. The accessibility to this data is key to allowing the work of an engineer press forward. The Consortium of Universities for the Advancement of Hydrologic Science (CUASHI) carried out an important project called the Hydrologic Information System (HIS) that outlines their plan to improve the problem of data inaccessibility. The World Water Project at Brigham Young University (BYU) is currently working with CUASHI to help advance solutions to “this problem through the development of tools, technologies, and standards for sharing water and climate date in a manner conducive to rapid scientific advancement” (World Water Project, n.d.). The objectives are to remove the barriers to data sharing and overcome cost and technology barriers to facilitate global water exchange.

One form of technology comes in the development of the Tethys Platform. This platform was created by Nathan Swain as part of the NSF-funded CI-WATER project and is used to develop and host water resources web applications otherwise known as “apps”. The reasoning behind Tethys was “to make the development of engaging, interactive web apps for water resources as easy as possible” (Swain, 2015). The Tethys Platform can be a gateway to creating a cloud-based water resource data infrastructure. This project demonstrates the creation of a data retrieval web app created on the Tethys Platform.
1.1 **Purpose**

The creation of this app included two primary objectives: 1. Demonstrate the level of programming expertise required to create and publish an app with Tethys, and 2. Create an app that can retrieve, plot and print the Dominican Republic data from a Hydroserver.

1.2 **Data Observations and Storage**

During February 2015, I visited the Dominican Republic under the supervision of Dr. E. James Nelson. We worked directly with the National Water Resources Institute (INDRHI) in Santo Domingo. INDRHI provided water level data values from 02-15-2015 to 03-29-15 for five different sensors. I received an electronic file with all these data values from INDRHI. A new HydroServer database was created specifically for these data values and hosted on the BYU servers to allow for a publicly accessible database. As shown in Figure 1-1, HIS-Central harvests the data from the HydroServer and “contains copies of metadata which facilitates searches” (*HydroServer*, 2015).

![Figure 1-1: Key Components of CUASHI-HIS (*HydroServer*, 2015)](image)

---

*Figure 1-1: Key Components of CUASHI-HIS (*HydroServer*, 2015)*
Data values stored in the database are retrieved using a variety of retrieval clients. The query of the database depends on user-specified variables. For example, one can specify the site location, hydrologic variable and dates for the desired data values. The query will then call the data values from the specified site database using HIS-Central.

A new search client will retrieve the Dominican Republic data that is hosted on a HydroServer and harvested in HIS-Central. Using the Tethys platform a new client can be created in the form of a web application. The app will be able to query HIS-Central for the data values return a plot and file of the data values.
2 TETHYS PLATFORM

Tethys is a platform that was developed at BYU by Nathan Swain. This platform is used to create water resource web applications known as “apps”. “Tethys web apps are developed using a Python Software development kit (SDK) which includes programmatic links to each software component while being powered by the Django Python web framework” (Swain, 2015). A conceptual diagram and overview of the platform is shown in Figure 2-1.

Figure 2-1: Tethys Software Platform (Swain, 2015)
2.1 Tethys Gizmos and Templates

Tethys Platform has the objective “to make development of engaging, interactive web apps for water resources as easy as possible” (Swain, 2015). The hope is to enable the creation of a vast library of water resource apps. A Tethys Portal with several contemplated apps is shown in Figure 2-2. Tethys tutorials help provide understanding how to work with the app file structure, view, template, forms and other related items. These tutorials teach a user how to create a new Tethys app (http://tethys-platform.readthedocs.org/en/1.0.0/getting_started.html).

![Tethys Apps Library](image)

**Figure 2-2: Tethys Apps Library (Swain, 2015)**

Tethys makes it easier to create these apps with Gizmos and Templates. Gizmos are provided which are user interface elements that can be integrated into new apps with very minimal lines of code. In Figure 2-3, the date picker gizmo is shown as an element on the page. The code required is a simple python script with various options and the HTML referring to the
date picker. A programmer can browse the gizmo library and insert any gizmo in their app just by copying the associated code. The gizmo library provides gizmos such as buttons, range sliders, text input boxes, map views, plot views and many other useful user interface elements. The advantage of incorporating pre-coded gizmos allows for a less experienced programmer to easily create interface and visualization elements for an app.

![Date Picker]

**Figure 2-3: User Interface Elements known as “Gizmos” (Swain, 2015)**

The Django template system is “a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app” (Django, n.d.). The system is used to build the pages of the app and requires less raw HTML development.
Similar to the gizmos, this also facilitates the creation of an app for a less experienced programmer.

The Tethys app library demonstrates several apps and their functionality using the Tethys Platform. The example shown Figure 2-4 is the “Canned GSSHSA” app. Thousands of GSSHSA analyses were previously performed and stored in a database. The purpose of the app is to display the individual GSSHSA analysis that most likely represents the selected scenario.

**Figure 2-4: Canned GSSHSA Tethys App (Swain, 2015)**

The creation of this app took advantage of the usefulness and functionality of the Tethys Gizmos. All of the sliders and plots displayed are from the library of Tethys Gizmos. This app is one example of how Tethys Platform can be used and how the difficulty of programming and its barrier has been lowered.
3 APP DEVELOPMENT

This project involved using the Tethys Platform to develop a web app that retrieves data from a HydroServer stored on HIS-Central. In this section, I will explain my programming background and the experience I had while learning how to use the Tethys Platform. I expound on both the information and the misperceptions that were key to during my development of a new app. My experience will be used to help future programmer with their usage of the Tethys Platform and development of apps.

3.1 Programming Background

My education at Brigham Young University includes a Bachelor’s and Master’s degree in Civil Engineering. A relevant course during my Bachelor’s degree was CE En 270 Computational Methods. This class introduced me to the world of computer programming and taught me Visual Basic.

Over the past year with my Master’s degree, my coursework has expanded my computer programming knowledge immensely. The courses and their associated programming languages are as follows:

- CE En 414: Intro to GIS
  - Python
- CE EN 594R: Hydro-informatics
These courses were introductory, but have allowed me to program using each of the programming languages. In Table 3-1, I have rated my level of skill with each of the programming languages.

Table 3-1: Programming Language Skill Levels

<table>
<thead>
<tr>
<th>Programming Language</th>
<th>Level of Skill (Scale 1 to 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual Basic</td>
<td>2</td>
</tr>
<tr>
<td>Python</td>
<td>3</td>
</tr>
<tr>
<td>HTML</td>
<td>2</td>
</tr>
<tr>
<td>JavaScript</td>
<td>2</td>
</tr>
</tbody>
</table>

I tested to see if this level of programming skill is sufficient for a “typical” user for the Tethys Platform. The development of my app helped show that a user with a general programming background could successfully develop and publish an app using the Tethys Platform.

3.2 Install Tethys Platform

The first step was to install Tethys Platform (See Tutorial: http://tethys-platform.readthedocs.org/en/1.0.0/installation.html). The tutorial provided clear instructions and graphics for the full installation process. I installed the Linux version of Tethys Platform on my MacBook Air. I used Parallels as my virtual machine application. I did not experience any problems with installation or processing speeds during my use of Tethys. From the available versions (Windows, Mac, & Linux), I used the Linux version since I was familiar with the Linux terminal.
3.3 **Perform the Tethys Tutorials**

The app development began with the Tethys tutorials to understand how to create an app on the platform. The tutorials provide instructions regarding the design of the visual representation of the app and modify the functions the app performs. The development pattern used to develop Tethys apps is called “Model View Controller” (Swain, 2015). It is defined as “the Model represents the data of the app, the View is composed of the representation of the data, and the Controller consists of the logic needed to prepare the data from the Model for the View and any other logic your app needs” (Swain, 2015). The tutorials help the user become familiar with the files that are associated with each part of the development. When I was faced with most challenges, I could refer back to the tutorials and find the explanation to my misperception. The tutorials can be found at [http://tethys-platform.readthedocs.org/en/1.0.0/getting_started.html](http://tethys-platform.readthedocs.org/en/1.0.0/getting_started.html). The following is an outline of the key parts of the tutorials that benefited me in my app development.

The tutorials begin with the initiation of the new app project by using a scaffold. The scaffold “generates a Tethys app project with minimum files and the folder structure that is required. Initially it was challenging to understand the folder structure and how the python files worked with the HTML and JavaScript files. Here are three key files found within the “Model View Controller” folder structure.

**File: app.py (Model)**

This is the app configuration file. This file contains the app class that is used to configure the app such as the name and color. The variables defined for my app class are shown in Figure 3-1. The other portion of this file adds the controllers used for the app. I added two controllers
for my app; one was for the home page and the other for my plot page. The layout and functions of those pages are designed in the next two files.

```python
name = 'Observed Hydrologic Data'
index = 'observed_data:home'
icon = 'observed_data/images/logo.png'
package = 'observed_data'
root_url = 'observed-data'
color = '#e67e22'

url Maps = (UrlMap(name='home',
                    url='observed-data',
                    controller='observed_data.controllers.home'),
          UrlMap(name='plot',
                 url='observed-data/plot',
                 controller='observed_data.controllers.plot'))

```

**Figure 3-1: app.py file portions used for the Observed Data App**

**Files: home.html (View)**

The app templates are created using base.html, home.html and any other created pages. The templates use Django Template Language. The block tags of the home.html (shown in Figure 3-2) template correspond with different parts of the app interface (shown in 3-3).

Part #1 is the “app_navigation_items”. In this block, you can specify what is put in the navigation tab on the left side. The class is specified along with any app URL.

Part #2 is the “app_content”. This block allows you to place text and gizmos for the content of your app. I used the google_map_view gizmo for my map and have the options called “map_options”. The map options are specified in the controller.py file. I also created a form on this page and place some select_input and date_picker gizmos. I placed these gizmos in the form in order to send the inputted data in these gizmos from the home controller to the plot controller. I did not know how to send variables between controllers until I was shown how to create a form. The tags of `<form> </form>` include any content placed between the tags.

Part #3 is the “app_actions”. This block is generally used to place any action you wish your app to perform. I used a submit button gizmo to submit my form created in the “app_content” block. I specified the URL of the controller that will use the inputted variables.
**File:** controller.py (Controller)

The controller coordinates between the view and the model. Often this means querying a database and transforming the data to a format that the view expects it to be in. The Controller also handles most of the application logic such as processing and validating form data or launching model runs. In a Tethys app, controllers are simple python functions.

When creating a new controller, it will need to be associated it with a URL by creating URL map for it. When a URL is requested, the controller that it is mapped to will be executed.
This URL is added to the app.py file as described above. Additionally, a new HTML file for the controller template will need to be created and placed in the templates folder.

After the first tutorial is complete, there are two comprehensive tutorials that will act as reviews of “Model View Controller” development in Tethys Platform. I found the advanced concepts and user input and form tutorials to be very helpful to use as references.

3.4 Library of Gizmos

Gizmos are modular user interface elements that can be inserted into your app with few lines of code provided by the library of gizmos. Within this library are a wide variety of gizmos that can be used within your app development. I recommend becoming familiar with all of the available gizmos during the planning stage of the app development. This will help with the app design and plan to know what is and isn’t possible with the current set of gizmos.

Figure 3-4: Tethys Developer Tools (Swain, 2015)
Each gizmo has provided python script for your controllers.py file and Django Template script for your HTML file. It would also be helpful to understand the different options for each gizmo as well. Gizmos have many useful options and also some limitations. For example, the google_map_view gizmo options do not provide a way set the center or the zoom level of the map, but do provide a legend. While the map_view gizmo does allow you to set the center and zoom level, but does not provide a legend. Understanding the gizmos will help simplify the app development process.

3.5 **Create the New App Scaffold**

After the Tethys installation, tutorials and gizmo tour, you will be ready to start developing your own Tethys app. It begins with creating a new app file scaffold, as done with the first tutorial. After this is done, your app is a blank canvas and ready to begin designing. The design process will follow the steps outlined in the tutorials and you will apply your understanding of the “Model View Controller” development pattern.

3.6 **Begin New App Development**

Now that the new app scaffold is set up, we are able to see the app on the app library page. This app now all the parts of the app interface. After looking over the gizmos and planning my app layout, I first started with creating a new page to act as my plot page. I was then able to begin placing all of my gizmos on the home page or plot page. The gizmos acted as static placeholders that would soon be associated with any functions that were necessary for my app. I would recommend beginning with the placement of gizmos, which will help you understand what is needed to properly use the gizmo. I will take you through my app development with using the gizmos and assigning functions to the gizmo.
The map was created using the editable map gizmo. The sites that are placed on the map are retrieved from HIS-Central using the bounding box sites web service. Any site within the specified bounding box will be put into the list of sites posted on the map. The site list was difficult to create because of the required format for the map gizmo. The sites that were retrieved from HIS-Central were parsed and placed in an appropriately formatted site list. The map will also create a legend with the site name and arrow color of the site. A script was written to zoom to the Dominican Republic to allow for the viewing of the sites. (See Figure 3-5)

![Figure 3-5: Map of the Observed Data App](image)

The input form is a group of gizmos to query the HydroServer. The first gizmo was a select input gizmo where the gizmo is populated by a specified list. The site input is populated by the site list described above. Working with the site list was difficult because the select input gizmo had a required list format. The sites were parsed and placed in an appropriately formatted site list (site name, number of site). The benefit of the gizmo list format was known and could be produced. The variable input allows the user to choose what type of hydrologic data retrieved from HIS-Central. The last two input gizmos are date pickers. The user specifies the desired start and end date for the retrieved data values (See Figure 3-6).
The submit button sends a request to HIS-Central with the input variable to retrieve the associated data values. This part of the app was the most difficult because I had to understand how to properly communicate with HIS-Central and my HydroServer. The process initially seemed simple but I eventually realized the complexity of the task. Using many sources for help, eventually this function was created to take the input variables and retrieve the data values for the plot page.

The plot page used the plot gizmo to visualize the retrieved time series data from HIS-Central. The retrieved data had to be parsed and placed in a properly formatted list of (data value, date time). Using the following input variables of Site: Paso de Lima, Variable: Stage, StartDate: 2015-02-15 and EndDate: 2015-03-10, the plot was created shown in Figure 3-7.

The data values shown in the plot could also be downloaded as a .csv file. The download data button shown in Figure 3-7 was created using a button gizmo. The button was coded to parse the retrieved data values with the date/time into a comma-separated values file. The file is stored temporarily on the server in a folder named after the username. This temporary storage allows for multiple users to use the app without any errors. The data values file is the final output showing the successful retrieval of data from a HydroServer on HIS-Central.
3.7 GitHub Repository

My code used to develop the Observed Data App can be found in this GitHub repository: https://github.com/CI-WATER/tethysapp-observed_data Among the files in this repository is the HTML script that is used as the app layout template and the python script that queries HIS-Central for time series data. This repository will act as an additional guide and reference for the development of future Tethys apps.

3.8 Tethys App Demo Server

The Observed Data app is currently on the demo server for Tethys apps. Accessing this demo via http://demo.tethys.ci-water.org will allow you interact with the app. It will let you observe the Telemetric sensors locations on a map and also select a site, variable, start and end date to see the results on a plot.
4 PROJECT OUTCOME

This project demonstrated the creation of a web app that retrieves data from a HydroServer stored on HIS-Central. It was focused on providing an example that the Tethys Platform has lowered the barrier of programming and app development. While my app demonstrates how data can be retrieved from CUAHSI HIS, this project serves as both an experience and a guide for future developers with how Tethys Apps can be developed.

4.1 Tethys Platform Outlook

I spent an estimated time of 25 hours with the installation, tutorials and reviewing the platform. The HydroServer and HIS-Central setup took an estimated 25 hours of work. And lastly, the development of the app required an estimated 100 hours. I believe that I wouldn’t have been able to develop a quality app without the Tethys Platform. I was able to fulfill my objective by creating an app with Tethys while having a general programming background.

While I did not exhaustively test and use Tethys, I still believe my experience was beneficial and my conclusion valid. After working and successfully making an app on the Tethys Platform, I endorse that it does create a more simple approach to water resource app development. Tethys Platform has successfully lowered the level of programming experience required to create and develop an app by its simple use of templates and gizmos. Tethys Platform can be a gateway to creating a cloud-based water resource data infrastructure.
4.2 **Observed Data App Outlook**

The project evolved to the publishing of the HydroServer on HIS-Central, which helped allow the app query more than just an individual HydroServer. While it is hopeful that the code and developed app can continue progressing, there are also several different uses for my app.

Future water resource apps could need the capability of retrieving time series data to act as an input parameter. My code could be used to retrieve input data directly from a specified database. This will simplify the app development process for the programmer.

Another use of the Observed Data app could be to simply retrieve data from a different database. Currently it is set to query HydroServer’s hosted at BYU, but could be programmed to query another database. After changing the database, comparison to any time series data stored in the database could be made in a quick and easy manner.

It is assumed that this app will primarily be used as an example and help for future app development. This paper helps outline the steps, obstacles and suggestions for the app development process on the Tethys Platform.
REFERENCES


