Week 1 - Goals and Objectives

Contents

Chapter 1

Week 1 - Goals and Objectives: Introduction

Internet mapping technologies are an important component of geospatial data capture, sharing, visualization, and delivery. This course provides a survey of current and emerging internet and geospatial interoperability standards, technologies, and capabilities. The emphasis of the work in this class will be hands-on experience in four critical aspects of Internet-enabled mapping:

- The basic concepts behind web development and web mapping technologies that enable the delivery of maps and mapped data through web browsers

- The Open Standards that facilitate the exchange of map images and geospatial data over the internet

- The use of published standards-based services in desktop mapping applications that implement those standards

- The deployment of standards-based geospatial map and data services that other systems and users may make use of

The specific class objectives that relate to these activities and departmental curriculum objectives for undergraduate and graduate students in the Geography Department include the following:

- Students will understand the concepts geospatial data and service interoperability

- Students will be able to define the specific requirements of a particular analysis or project and identify the interoperability standards that are capable of meeting those requirements

- Students will be knowledgeable in the core technologies that they may use to produce their own internet-enabled mapping capabilities

- Students will understand the strengths and limitations of current internet mapping technologies for generating cartographically effective map products.

The weekly goals, objectives and assignments are outlined below

Chapter 2

Introduction

This week we will review the content and structure for the course and spend some time getting to know each other. Following this we will spend some time setting up some of the tools that you will be using for the course in developing your portfolio of materials.

Chapter 3

Class Prep

- Wikipedia article - History of the World Wide web

- Web Development tutorials (for weeks 1 & 2)

  - *HTML: A Beginner's Tutorial by Chris Coremans*
    * Introduction
    * 1. Chapter 1: Getting Started
    * 2. Chapter 2: Working with Text
    * 3. Chapter 3: Links
    * 4. Chapter 4: Images
    * 5. Chapter 5: Lists
    * 6. Chapter 6: Tables
    * 7. Chapter 9: Styling with CSS
  - *An Introduction to Web in HTML & CSS, by Cassidy & Camryn Williams* - Video Tutorial (basic)
  - *Modern Web Development with HTML5 and CSS, by Rachel Andrew* - Video Tutorial (in-depth)
  - *Introduction to Javascript (Tutorial)*
    * 1. *Nuts 'n' Bolts*
    * 2. *The DOM*

- Version Control using Git and Github (for weeks 1 & 2)

  - Ferdinando Santacroce. (2017). *Git Essentials** - Second Edition. https://www.safaribooksonline.com/library/view/git-essentials-/9781787120723/
    * *Getting Started with Git*
    * *Git Fundamentals - Working Locally*
    * *Git Fundamentals - Working Remotely*
  - Peter Bell. (2014). *Git and GitHub LiveLessons—Workshop (Video Training)*. *https://www.safaribooksonline.com/and-github/9780133992748/*
    * Lessons 1, 2, 3, 7, and 8

Chapter 4

Reference Materials

Class Syllabus

Chapter 5

Weekly Milestone - Creating Your GitHub Repository and First Web Page

Developing content to go onto the web has evolved from a solitary effort to one where teams work together in developing components of larger web sites. These teams need to have a variety of tools to enable their work. Some of the most important tools enable code sharing with the team, and in projects based on the Open Source software model the rest of the world. The GitHub web platform uses the Git distributed version control system to enable sharing of code and hosting static web pages based on that shared code.

You will be using a public GitHub repository and associated *Project Pages* collection to build your class portfolio during the course. You will learn how version control operates, and how to provide comments and keep notes on your work and comment on the work of others (this will be part of our peer review process).

While the work we do this and next week will be directly through the editor integrated into the GitHub system, you will need to install a desktop application (such as the SourceTree application [recommended for the class], the GitHub Desktop application, or, for the more adventurous, one of the Git command line tools) that allows you to develop your web pages on your local computer and then update the files on the GitHub system when you want to share a new version.

For this milestone we will walk through the process of creating your repository in GitHub, creating your first web page, previewing that page on your local computer, changing the page, and updating the page on GitHub. For this milestone we will do this as a manual process which we will streamline in the coming weeks.

Create Your GitHub Account and Portfolio Repository

For your work in this class you will build your portfolio within your account - which you will need to create if you don't already have one.

The first step in the process of creating your portfolio is to create a new (or log into an existing) GitHub account, and create *repository* in GitHub within which you will put your portfolio materials for sharing within the class. Please follow the following steps to create your repository:

1. Go to the GitHub homepage and follow the onscreen instructions for creating a new account. If you already have an account you can skip this step. Once you have created your account you can login and continue the process of creating your class repository.

15

Figure 5.1: GitHub home page

Unless you have specific reasons to do otherwise select the default options for step 2 of the account creation process.

(optional) Complete the online survey from GitHub associated with your account.

2. Check your email inbox and verify your newly created GitHub account.

3. Select "Start a project" from the web page that is presented when you confirm your email address (or complete the account creation steps above).

4. Create your portfolio repository by choosing the following options in the page that is presented when you choose "Start a project" in step (3) above.

After your successfully create your `portfolio` repository you should see the home page for your new repo:

Make note of the web address of this page (even email it to yourself) to make it easy to get back here later - `https://github.com/<your username>/portfolio`

Respond to the "GitHub Username and Repository Survey" in the assignments section of Learn with the GitHub username and repository that you will be using for the class

Create Your First Web Page

To create your first web page within your portfolio repository you need to first enter your repository, add a new file, modify its contents, and commit your modifications back to the repository to save your changes.

1. Go to your `portfolio` home page - `https://github.com/<your username>/portfolio` - either by going directly to the link (above) for your repository or by selecting the repository from your account home page - `https://github.com/<your username>`.

2. On the page that comes up listing the files in your repository, click the "Create new file" button above the list of files.

Figure 5.2: GitHub setup step 2

Figure 5.3: GitHub online survey
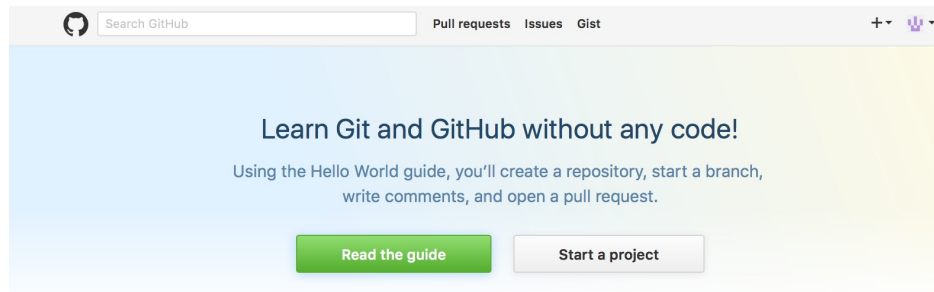
Figure 5.4: Start project page
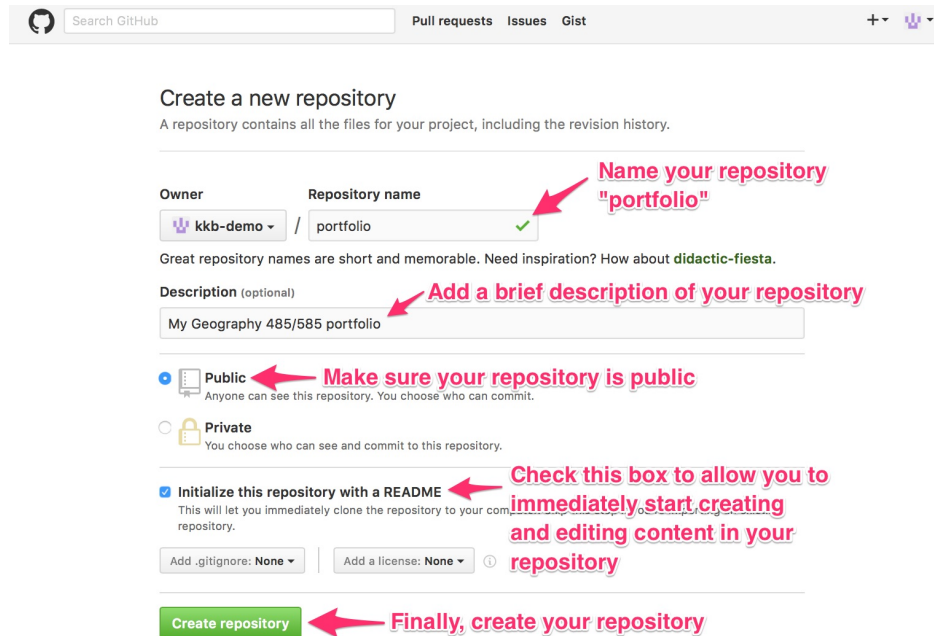
Figure 5.5: Create repository page
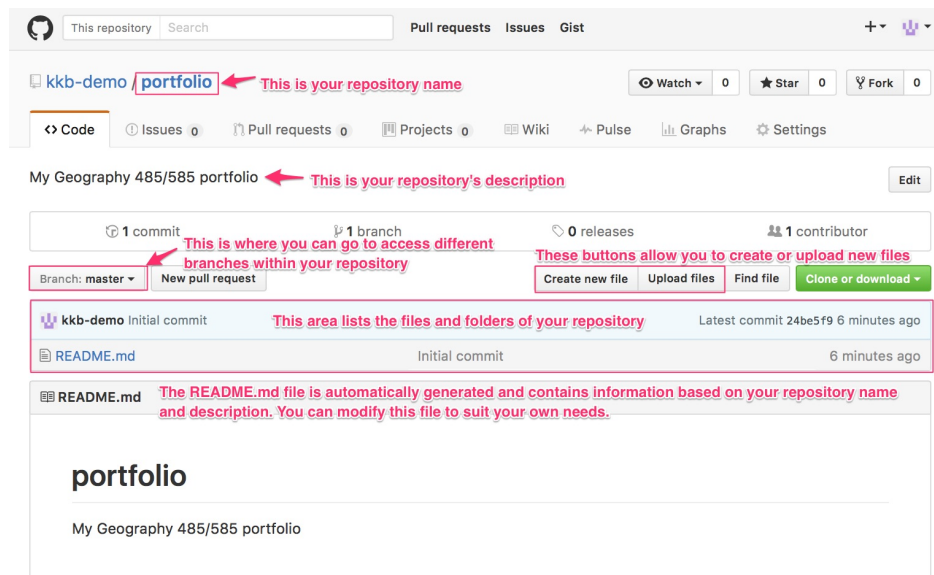
Figure 5.6: Repository home page
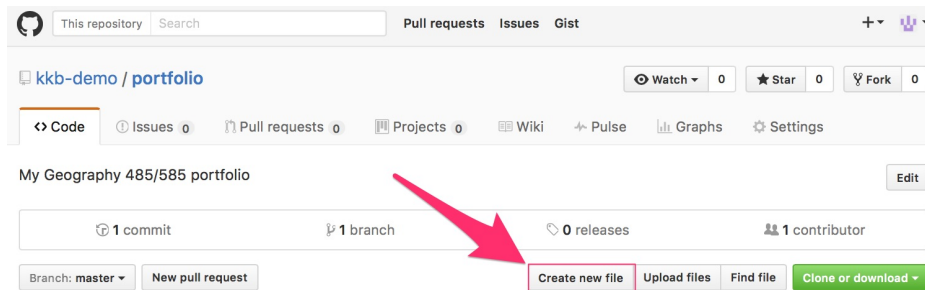
Figure 5.7: Create a new file

Which will take you to the editor for your new file.

3. Enter the name of the file ((1) on the figure below) that you are creating as "hello-world.html"

4. Enter the following text into the text entry area (2) under the filename field.

```
1  <html>
2      <head>
3      </head>
4      <body>
5          Hello World !!!
6      </body>
7  </html>
```

5. Add a brief comment (such as "Created hello-world.html from provided text") in the first field under the "Commit new file" title (3). You can optionally add a more detailed description in the next field (4) if you like.

6. Keep the default option to "Commit directly to the master branch"

7. Click the "Commit New File" button (5) to commit your change and save the file

Step 3 - Preview Your Web Page in a Browser

Since your repository in GitHub is public you can use GitHub to host your portfolio and its content on the web. GitHub's hosting capabilities are limited to static content (i.e. files directly accessible over the web), but this will meet our needs for the class very well. To enable GitHub's web hosting capabilities for your repository you need to change the `GitHub Pages` option in the settings for your repository. First, click the `Settings` button near the top of your repository home page:

Then modify the `GitHub Pages` setting to use the `Master` branch as the source for your GitHub pages web site. Click the `Save` button next to your update.

After saving your changed `GitHub Pages` setting you can view your web page using the following pattern:

`https://<your username>.github.io/<repository name>/<page name>`

which translates into the following (assuming that you followed the instructions above for naming your repository and file:

`https://<your username>.github.io/portfolio/htllo-world.html`

Using this approach you can modify and preview your site in real time as you commit your changes within GitHub. You can also preview your portfolio based on files on your local computer by installing a GitHub client onto your computer and `cloning` your GitHub repository to your local computer. This is done using
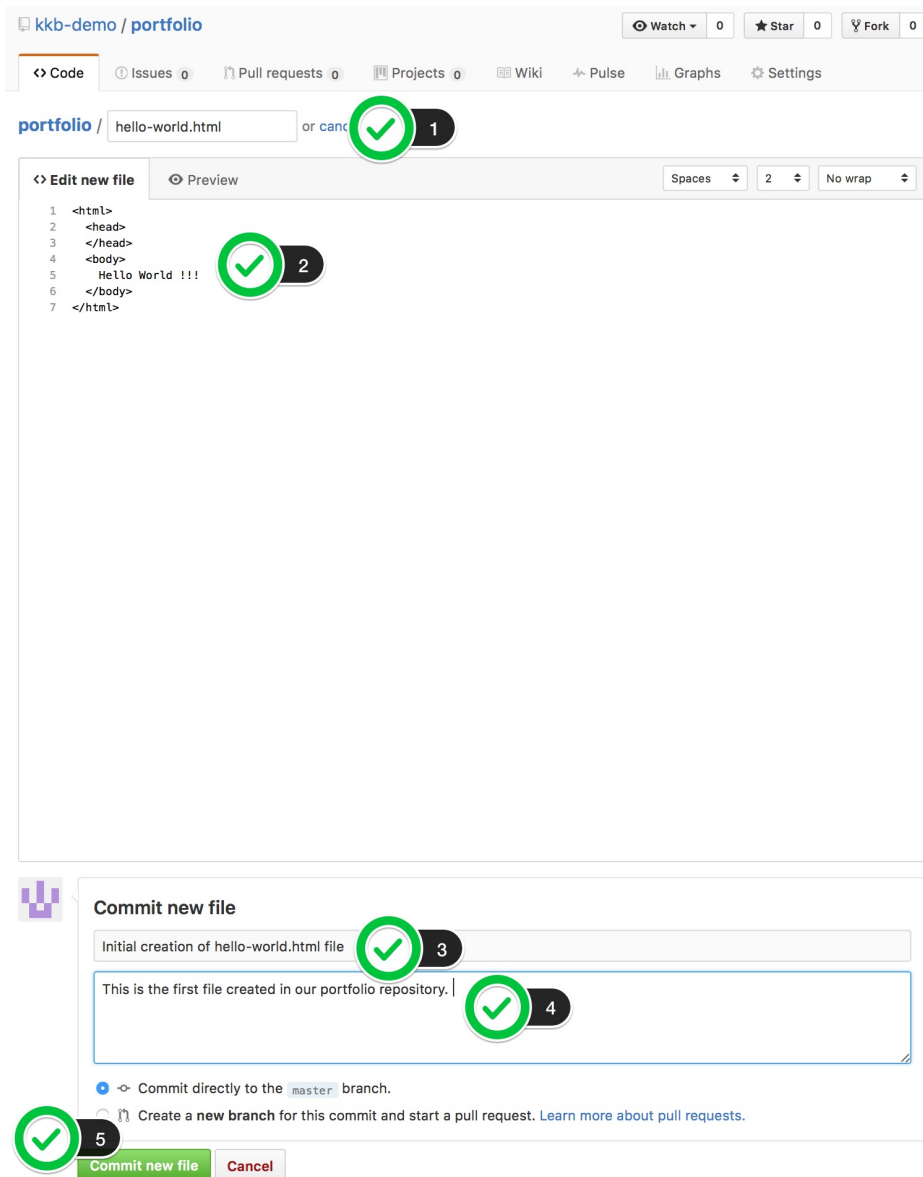
Figure 5.8: GitHub file creation/editor page
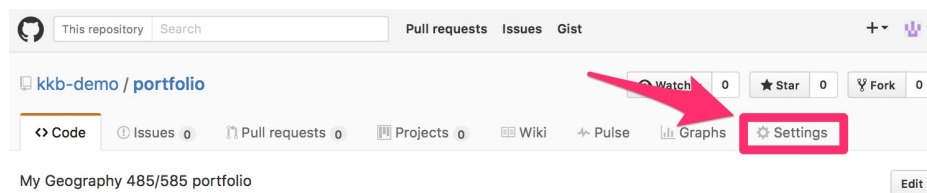
Figure 5.9: Settings button in GitHub
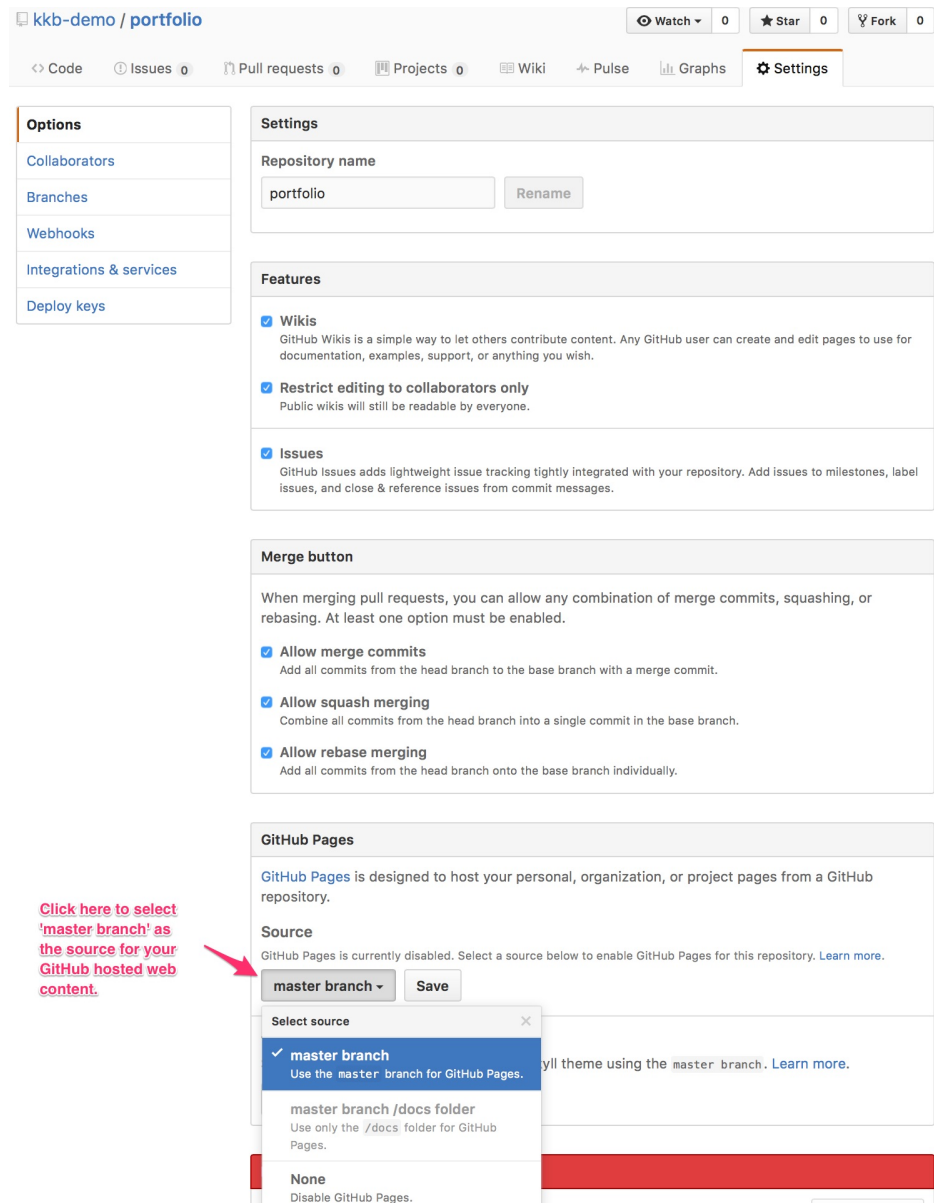
Figure 5.10: Update settings for GitHub web publishing

a desktop application such as *Sourcetree*, the *GitHub Desktop Client*, or another *Git* client appropriate to your operating system. Once you've cloned your repository you can work with the files (including previewing them) on your local computer and when ready `push` those files back into GitHub for online access, viewing and sharing. **While you can work on your files locally - peer review, troubleshooting, and grading will be based on the content in your public GitHub repository**.

Confirm that the display resembles something like the following:

6. If the page does not appear as you like, edit it on GitHub, commit your change and preview it again. Repeat until you get what you expect.
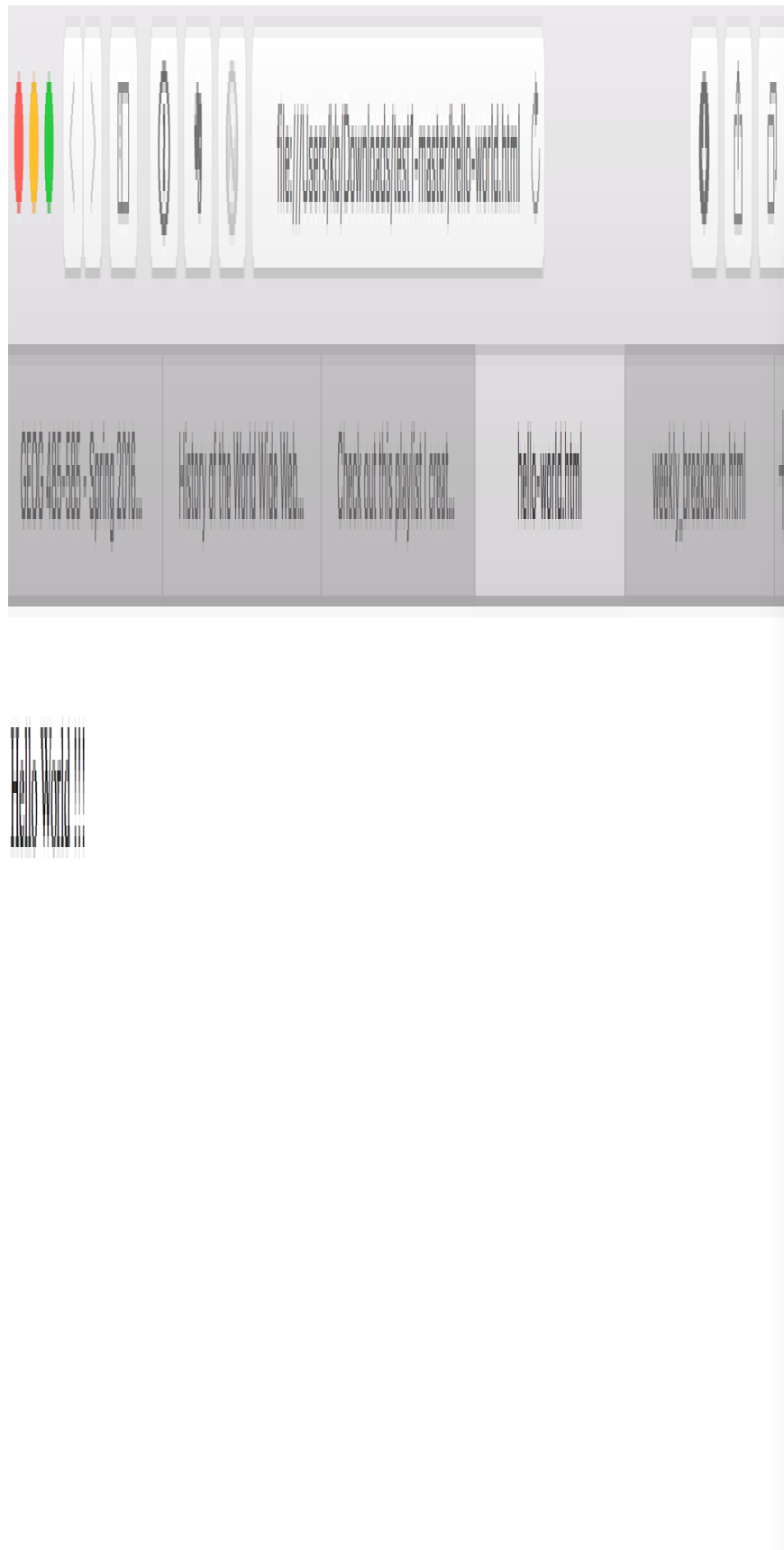
Figure 5.11: Sample `hello-world.html` file when viewed in a web browser

Chapter 6

Introduction

This week we will begin to build our foundation for developing material to be shared over the Internet via the World Wide Web. In particular we will cover the basic process of web development, define the parts of a web page, and spend some time learning about the different *languages* and define the key components of a web page: its structure, presentation, and behavior.

The presentation of information over the Internet is dependent upon the use of standards that have been developed for defining the *structure*, *presentation*, and *behavior* of content. This week we will begin working with the key technologies that define these three components of web content.

These concepts will be illustrated through reference to several simple web pages which are progressively modified to integrate all three of these components.

Expected Outcomes

By the end of this class module you should understand the following:

- The basic process of web development

- The parts of a web page

- The role of the three web page components: *structure*, *presentation*, and *behavior*

- Be able to write your own basic web page with your own content and make it available over the web

Key Concepts

- Parts of a web page

- Structure = X/HTML

- Presentation = CSS

- Behavior = Javascript

- Iterative Development

Chapter 7

Class Prep

- Complete readings and video tutorials from week 1.

Chapter 8

Weekly Milestone - Create a More Complex Web Page and Style It

This week's milestone activity takes you through the process of creating two more web pages in preparation for next week's work with the Google Maps API in developing your first web mapping page. These pages will be:

1. A *home page* for your portfolio that will be the access point for all of the materials you create, and

2. A variant of your *home page* that has been styled to modify its appearance.

Step 1 - Open the *home page* template in your web browser.

Step 2 - Copy the code in the home page template into a new file named `index.html` in your GitHub repository.

Step 3 - After you have committed the `index.html` file to your repository preview it in your browser to see what it looks like when accessed as a static web page. Verify that it looks the same as the preview of the template file as a web page.

Step 4 - Answer the questions in the Milestone 2 assignment in Learn

Step 5 - Flesh out the `index.html` page that you created above (*Step 2*) with information specific to you and your interests based upon the content areas in the page. After making sure that your `index.html` is in the same directory as the `hello-world.html` file you created in Milestone 1, add a *relative* link to your `hello-world.html` file to the "milestones" section of your `index.html` page by modifying the line

```
<p><a href="">Hello World</a></p>
```

to look like this

```
<p><a href="hello-world.html">Hello World</a></p>
```

Save your change and test it in the browser by clicking the link on your `index.html` page in the browser. If it successfully opens your `hello-world.html` page you have properly built your link.

Step 6 - Make a copy of your `index.html` page by copying the content of the page and pasting it into a new document in your repository named `index_styled.html`.

Experiment with some of the styling capabilities described in the W3Schools CSS Examples page (http://www.w3schools.com/css/css_examples.asp) on `index_styled.html` page you created above. Make at least three stylistic changes to the `index_styled.html` page. Add a link to your `index_styled.html` page to your home page (`index.html`) under the `milestones` section.

29

Chapter 9

Introduction

This week we will begin our work with the popular Google Maps *Application Programming Interface* (API) in developing an interactive web-based mapping client. This development activity will build upon the the work you've done over the last couple of weeks in developing basic web pages by using the capabilities that Google has made available for building mapping interfaces based upon their Maps platform. You will begin working with javascript as a client programming language to both interact with Google's servers and to provide the needed information for Google's mapping tool in your web page.

Expected Outcomes

By the end of this class module you should understand the following:

- What an Application Programming Interface (API) is

- How Javascript can be used to define the behavior of elements in a web page

- What the basic structure of a javascript code block for defining a Google Maps - enabled page looks like

- How to write a basic web page that includes an interactive Google Map

Key Concepts

- Application Programming Interface (API)

- Javascript and its location within an HTML page

- The interaction between javascript behaviors and structural elements in a web page

Chapter 10

Class Prep

- w3schools JavaScript Tutorial

- Google Maps API Tutorial

- Balkan Uraz, & Alper Dincer. (2013). *Google Maps JavaScript API Cookbook**. (Weeks 3 and 4)https://www.safaribooksonline.com/library/view/google-maps-javascript/9781849698825/

    - *Google Maps API Basics*
    - *Adding Raster Layers*
    - *Adding Vector Layers*
    - *Working with Controls*

- Google Maps API Key - a Google Maps API key is now required to write web pages that make use of the Google Maps API. You can get your API key here

Chapter 11

Reference Materials

Chapter 12

Weekly Milestone - Creation of a Web Page with an Embedded Google Map

In preparation for creating a web page with an embedded Google Map you should first answer the questions provided in Learn about what and how you want to map. As you define the type of map you want to build, think about a specific problem or topic that you would like to address with your map. In other words - *what is the story you want to tell with your map?*

In this exercise you will be generating the configuration for the base map (i.e. The Google Maps background layers). In future assignments you will add your own custom content to free-standing web pages that include a mapper based upon the base map you define here.

Once you have answered the questions in Learn about the map that you want to create, refer to the examples in the lecture notes, the Google Maps Tutorial, and this week's reading and video tutorial assignment to create a custom Google map.

Provide the link to your Google map in response to the last question in the assignment in Learn.

Chapter 13

Introduction

This week covers some additional topics related to the Google Maps API, particularly focusing on styling the Maps base maps using the Create Map Style wizard and integrating the javascript generated by the wizard into the base web page code developed last week; and, using Google's Fusion Tables tool to create and manage tabular data for mapping and other visualization. We complete our work with the Maps API with an example of a more "real" example of a maps-enabled web page.

Expected Outcomes

By the end of this class module you should be able to:

- Generate a Google Maps JSON style using the *Create Map Style Wizard*

- Integrate that JSON into your map client page for styled basemap display

You should also understand

- The potential of *Fusion Tables* as an alternative source of data to integrate into a custom Google Map page

- The potential structure of an *operational* web page, including the physical separation of page components (structure, presentation, behavior) into separate files

Key Concepts

- Generating Google Maps styles

- Integrating styles into a Google Maps page

- Fusion tables as a data source for Google Maps maps

- Separation of structure, presentation, and behavior in web development

Chapter 14

Class Prep

Continue with materials from last Week

Chapter 15

Reference Materials

- Google Maps API Tutorial
- Google Maps Styling Reference
- Google Maps Fusion Mapper

Chapter 16

Weekly Milestone - Styling of an Embedded Google Map

Make a free-standing web page based upon the Google Map that you created as part of last week's milestone. Use the Google Create Map Style wizard to define *at least* three modified base map styles and integrate the JSON generated by the wizard into your new Google Map page.

Chapter 17

Deep Dive - Creation of a Google Maps Web Page with Custom Points and Labels

In your milestone for this week you built a styled Google Maps base map for a particular region of interest. For this *deep dive* assignment create a new free-standing web page that includes a brief description of the topical focus of your mapper:

- The type of information that you want to depict in your map
- Your reasons for selecting the specific area shown in the map
- A description of what you are trying to communicate with the map

Embed the base map that you initially created for your milestone into this new web page.

- Add 5 overlay objects to the map that relate to specific items of interest or importance. These overlay objects may be *markers*, *polylines*, or *polygons*. Make sure to include descriptive titles for each object.
- Add an *infobox* to each object that contains additional detailed information about the object

Chapter 18

Introduction

Core the the development of distributed mapping systems over the internet is the concept of web services and the interoperability upon which they are based as the means of communication between systems. This week's lecture and focuses on the core concepts of geospatial *Services Oriented Architectures* and the open interoperability standards from the *Open Geospatial Consortium* that enable the exchange of map images and data over the web.

Expected Outcomes

By the end of this class module you should understand the following:

- The difference between raster and vector data formats and strategies for retrieving information about supported file formats

- The three general tiers of a geospatial services oriented architecture and the components that may exist in those tiers

- The key Open Geospatial Consortium standards for access, data, and representation

Key Concepts

- Raster and Vector Data Models

- The tiers of a geospatial services oriented architecture

- The constituent components of SOA tiers

- The role of OGC services in providing connectivity between SOA tiers

- The OGC WMS, WFS, WCS, GML, and KML standards and their respective capabilities and purposes

Chapter 19

Class Prep

1. Yang C, Raskin R, Goodchild M, Gahegan M. Geospatial Cyberinfrastructure: Past, present and future. Computers, Environment and Urban Systems. 2010;34: 264–277. doi:10.1016/j.compenvurbsys.2010.04.001 http://www.sciencedirect.com.libproxy.unm.edu/science/article/pii/S0198971510000268

2. Granell C, Díaz L, Gould M. Service-oriented applications for environmental models: Reusable geospatial services. Environmental Modelling & Software. 2010;25: 182–198. doi:10.1016/j.envsoft.2009.08.005 http://www.sciencedirect.com.libproxy.unm.edu/science/article/pii/S1364815209002047

3. Foster I. Service-Oriented Science. Science. 2005;308: 814–817. doi:10.1126/science.1110411 http://science.sciencemag.org.libproxy.unm.edu/content/308/5723/814

Chapter 20

Reference Materials

None

Chapter 21

Weekly Milestone - Fun with data

Question 1 Define a data theme that you would like to focus on for this assignment

Download three data products from one or more of the following online data repositories or another data repository that has data that interest you.

- New Mexico Resource Geographic Information System
- The US National Map Data Download Site
- NOAA's National Climate Data Center *Climate data online: Data discovery* site
- US Census Bureau - Geography - TIGER Data

Question 2 For each of the three datasets provide the following information

- The name of the dataset
- The filename(s) for the dataset
- A short (1-2 sentence) description of the dataset's contents
- The bounding box (provided as the minimum and maximum extent in the N-S and E-W directions) in the **native units and coordinate system**
- The coordinate reference system - by **name and EPSG code**

Chapter 22

Introduction

This week's class focuses on three open interoperability standards that are the most broadly used of the standards that we will be covering.

Extensible Markup Language (XML) The World Wide Web Consortium (W3C) standard that is the foundation for many other service and data standards including: the service metadata (GetCapabilities) for the OGC WMS, WFS, and WCS, Geography Markup Language (GML), and KML.

KML Formerly known as Keyhole Markup Language, an OGC standard since 2008, KML is a combined geospatial data and representation standard that enables the combined transfer of both location-based data and styling information within a defined XML model.

Web Map Service (WMS) The OGC standard for providing on-demand map visualizations based upon user provided parameters reflecting selected data layers, defined areas of interest, image formats, and optionally time of interest.

Expected Outcomes

At the end of this class students should have an understanding of the following:

- The basic characteristics of XML documents, including the concepts of *well-formed* and *valid* XML

- The capabilities of KML for providing both data and representation information for geospatially referenced data.

- The request-response model for OGC WMS, including the required and optional request parameters for the *GetCapabilities*, *GetMap*, and *GetFeatureInfo* requests; and the response types generated in response to those requests.

- A general familiarity with the linkage between the WMS and KML standards

Key Concepts

- XML as a general standard for structured data exchange, with DTDs and Schemas defining application specific data models

- KML as a data and representation standard for delivery of geospatial data and symbolization information into client applications, both desktop and web-based.

- WMS as a geospatial data visualization standard for providing online access to map images in a variety of formats for integration into desktop and web-based mapping applications

Chapter 23

Class Prep

- OGC Workshop White Paper

Chapter 24

Reference Materials

- OGC WMS Implementation Specification Version 1.0 - 2000, Version 1.1 - 2001, Version 1.1.1 - 2002, *Version 1.3.0 - 2006*

- OGC KML Version 2.2 - 2008, *Version 2.3 - 2015*

- Google Code KML Documentation

Chapter 25

Weekly Milestone - WMS

There are a large number of WMS services available on the web. One way to find interesting services is to search for them using standard search engines such as Google. Try searching for the following search phrase:

`"REQUEST=GetCapabilities"` and `"SERVICE=WMS"`

as a single search phrase

Question 1 What search engine did you use?

Question 2 How many 'hits' did you get?

Question 3 How useful (generally in terms of getting a pointers to live WMS services [defined as a *functioning* GetCapabilities request]) were the 'hits'?

Pick two of the services that included live "GetCapabilities" requests that you found above, and answer the following questions about each.

Question 4 (service #1) What is the URL for the full GetCapabilities request to the service?

What is the Name of the service?

What Format(s) are available for GetMap requests from the service?

How many layers are included in the service (*including nesting/nested layers*)?

Question 4 (service #2) What is the URL for the full GetCapabilities request to the service?

What is the Name of the service?

What Format(s) are available for GetMap requests from the service?

How many layers are included in the service (*including nesting/nested layers*)?

Question 5: For one of the layers in the first service, What is the name of the layer?

What is the SRS of the layer?

What is the name of the projection that matches the SRS EPSG code?

What is the LatLonBoundingBox of the layer?

Open the following GetCapabilities request in your browser. Select "View Source" from the browser menu to see the delivered XML document (it may appear as an unformatted string of text by default in your browser - if that is the case, save the file to your hard drive and view it in a text editor). Use the information in the XML capabilities document to formulate `GetMap` requests for the following map images. Include the requestsin your write-up. Comment on anything unusual that you notice in the images that are returned.

http://gstore.unm.edu/apps/rgis/datasets/92403ebf-aec5-404b-ae8a-6db41f388737/services/ogc/wms? SERVICE=wms&REQUEST=GetCapabilities&VERSION=1.1.1

Question 6 for the area surrounding Bernalillo County (-107.2,34.7,-106,35.25; EPSG:4326) for the
`g_2007fe_35_county` layer as a 200x200 pixel JPEG

for the same area and layer as a 500x500 pixel PNG

Chapter 26

Deep Dive - OGC Service Concepts

Question 1 What request type is common across all three (WMS, WFS, WCS) OGC web services that we have learned about?

Answer the following questions about a *WMS GetCapabilities* request

Question 2 What are the required parameters, and what do they represent?

What is returned in response to a WMS GetCapabilities request?

Answer the following questions about a *WMS GetMap* request

Question 3 What are the required parameters, and what do they represent?

What is returned in response to a WMS GetMap request?

What is the significance of transparency in WMS requests?

Question 4 What OGC request would you use to inform the configuration of a client application (like ArcGIS or QGIS) about an OGC service that you want to add layers from?

Which OGC request would you submit under the following circumstances (*include both the service type* [e.g. WMS, WFS, WCS], and the *request* [e.g. GetMap, GetCapabilities, GetCoverage, etc.] in your answer)

Question 5 You want a map image representing three layers of data in a single JPEG for a specified area of interest.

You want to retrieve data representing geometries and associated attributes for a road network, with the returned data in GML.

You want to retrieve data representing a digital elevation model (a raster dataset) in the form of a GeoTIFF.

Question 6 - What are the EPSG codes of the following Spatial Reference Systems WGS 84 (Geodetic CRS [geographic 2d])

NAD83 / UTM zone 13N

NAD27 / UTM zone 13N

Retrieve the GetCapabilities XML response from the following WMS, and answer the following questions.

http://gstore.unm.edu/apps/rgis/datasets/715663ba-c1c3-414c-84a7-c671526f8316/services/ogc/wms?
SERVICE=wms&REQUEST=GetCapabilities&VERSION=1.1.1

Question 7 What is the Title of the service?

Who is the Contact Person for questions about the service?

What are the available image formats for the GetMap request for this service?

What are the SRS/CRS's for which layers from this service are available (remember that nested layers inherit the SRS/CRS of their parent layers).

Question 8 Formulate a GetMap request for the "tl_2010_35_bg10" layer from this service, for a 500x500 pixel map image that is 0.05-degrees wide and 0.05-degress high, with the SW corner of the map image located at 35°N and -106°45'E (EPSG:4326). Include in your write-up the complete GetMap request.

Chapter 27

Introduction

This week's class focuses on two other key Open Geospatial Consortium standards that were created to enable access to geospatial *data* of a variety of types.

Web Feature Service (WFS) A standard designed for providing on-demand access to features (typically points, lines, and polygons and more complex combinations of these feature types) and their associated attributes, in a variety of formats, and optionally filtered by spatial and other query parameters.

Web Coverage Services (WCS) A standard focused on providing access to *coverages* representing a variety of data types, but particularly optimized for dynamic delivery of data based upon multidimensional gridded data.

Expected Outcomes

At the end of this class students should have an understanding of the following:

- The request-response model for OGC WFS, including the required and optional request parameters for the *GetCapabilities*, *DescribeFeatureType*, and *GetFeature* requests; and the response types generated in response to those requests.

- The request-response model for OGC WCS, including the required and optional request parameters for the *GetCapabilities*, *DescribeCoverage*, and *GetCoverage* requests; and the response types generated in response to those requests.

- An understanding of the distinction between WMS, WFS, and WCS for use in different usage scenarios (e.g. map images, vector, raster data access)

Key Concepts

- OGC WFS as a data access standard for *features* and their attributes with support for a variety of query methods including spatial and parameter values.

- OGC WCS as a data access standard for *coverages* representing spatio-temporal gridded data represented in 1- or more dimensions.

Chapter 28

Class Prep

Chapter 29

Reference Materials

- [OGC WFS Implementation Specification][http://www.opengeospatial.org/standards/wfs]
- [OGC WCS Implementation Specification][http://www.opengeospatial.org/standards/wcs]

Chapter 30

Weekly Milestone - WMS GetMap Requests, Map Scale and Aspect Ratio Calculations

You might have noticed in the WMS requests that you generated in the previous lab returned images that didn't look "quite right" relative to what you may know of the shape of familiar features.

For example, a WMS request for a 200x200 pixel PNG file (Figure 1) for an area surrounding Bernalillo County (-107.2,34.7,-106,35.25) from the previous lab would be (link):

```
http://gstore.unm.edu/apps/rgis/datasets/92403ebf-aec5-404b-ae8a-6db41f388737/
services/ogc/wms?VERSION=1.1.1&SERVICE=WMS&REQUEST=GetMap&BBOX=-107.2,34.7,
-106,35.25&LAYERS=g_2007fe_35_county&FORMAT=image/png&TRANSPARENT=TRUE&
STYLES=&SRS=EPSG:4326&WIDTH=200&HEIGHT=200
```

this request results in a map image that does not agree with the standard shape of Bernalillo county (depicted in QGIS - Figure 2) that we are accustomed to, regardless of the specific map projection being used.

This discrepancy is the result of a difference in the aspect ratio of the requested BBOX (-107.2,34.7,-106,35.25) and the requested image dimensions (200x200 pixels). *When you compose a WMS GetMap request, you need to make sure that the aspect ratio of both the image size and BBOX match.*

For example, if we calculate the aspect ratio of the BBOX we obtain the following values (remember that the BBOX is specified as a comma separated list of x,y coordinates: minx,miny,maxx,maxy):

$BBOX_{width} = x_{max} - x_{min} = (-106) - (-107.2) = 1.2°$

$BBOX_{height} = y_{max} - y_{min} = 35.25 - 34.7 = 0.55°$

$BBOX_{aspect-ratio} = BBOX_{width}/BBOX_{height} = 1.2°/0.55° = 2.1818$

If we want to retrieve a map image that is 200 pixels wide, we need to calculate an image height that yields an aspect ratio that matches the BBOX aspect ratio. Harking back to basic algebra:

$Image_{width} = 200px$

$Image_{aspect-ratio} = Image_{width}/Image_{height} = 200px/Image_{height} = 2.1818$

$Image_{height} = Image_{width}/aspect - ratio = 200px/2.1818 = 91.667px$

So, if we request an image that is 200x92 (we have to request pixel dimensions in integers, so rounding to the nearest integer) we should get a representation that closely approximates the proper shape of features. The modified WMS request with the new image's size is the following (link):

Figure 30.1: Returned map image for the region surrounding Bernalillo County for a WMS request with `BBOX=-107.2,34.7,-106,35.25`, `WIDTH=200` and `HEIGHT=200`

Figure 30.2: Area around Bernalillo County as viewed in QGIS based on the OGC WCS based on the same data source used for the WMS request illustrated above}
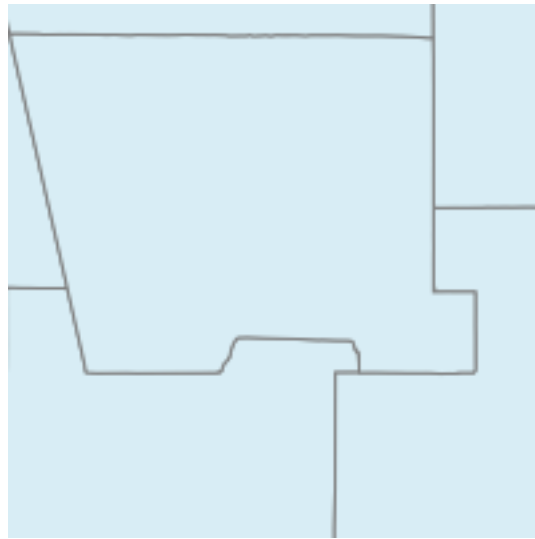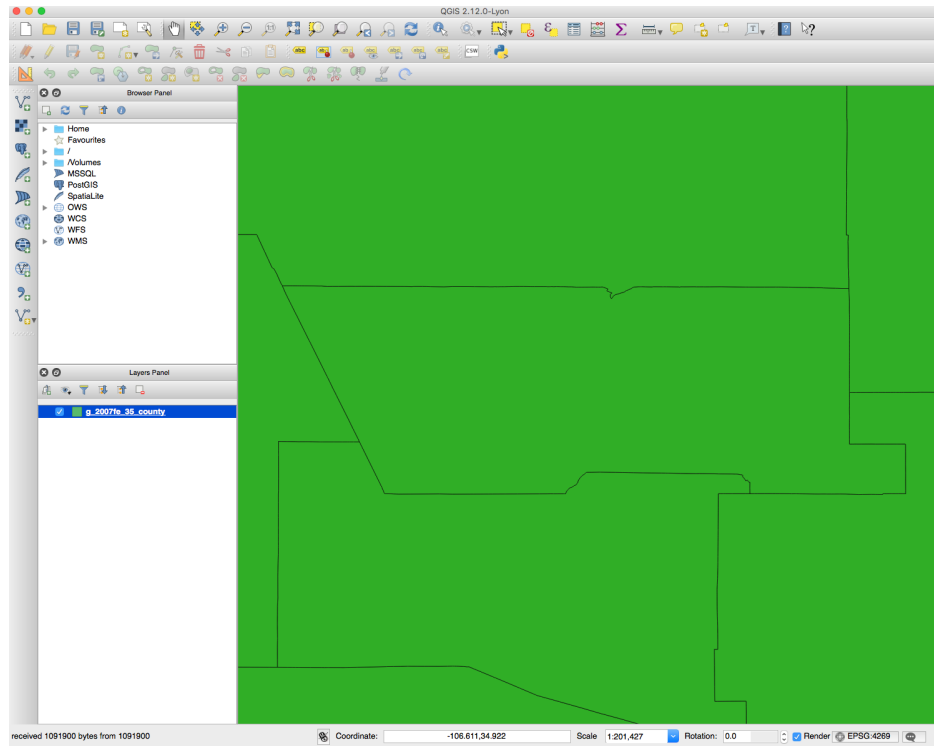
Figure 30.3: Returned map image for the region surrounding Bernalillo County for a WMS request with `BBOX=-107.2,34.7,-106,35.25`, `WIDTH=200` and `HEIGHT=92`

```
http://gstore.unm.edu/apps/rgis/datasets/92403ebf-aec5-404b-ae8a-6db41f388737/
services/ogc/wms?VERSION=1.1.1&SERVICE=WMS&REQUEST=GetMap&BBOX=-107.2,34.7,-106,35.25&
LAYERS=g_2007fe_35_county&FORMAT=image/png&TRANSPARENT=TRUE&STYLES=&
SRS=EPSG:4326&WIDTH=200&HEIGHT=92
```

This process may be reversed to request images of a fixed size for use in a client interface, with the requested BBOX modified to match the aspect ratio of the target image. If, for example, images are being requested for a client interface with a fixed map size of 600x400 pixels, a corresponding BBOX can be derived using the same calculation.

If, for example, the area of interest for a map is 2 degrees wide, we can calculate the target height (in degrees) using the aspect ratio of the desired image.

$Image_{aspect-ratio} = Image_{width}/Image_{height} = (600px)/(400px) = 1.5$

$BBOX_{aspect-ratio} = BBOX_{width}/BBOX_{height} = 2°/BBOX_{height} = 1.5$

$BBOX_{height} = BBOX_{width}/BBOX_{aspect-ratio} = 2°/1.5 = 1.3333°$

If our area of interest extends from -106 to -108 degrees East Longitude, we can use the known target height of 1.3333 to generate a WMS BBOX of the appropriate aspect ratio. If the minimum Latitude of interest is 34.7 degrees North Latitude, the maximum BBOX Y value would be

$y_{max} = y_{min} + BBOX_{height} = 34.7° + 1.3333 = 36.0333$

This set of calculations may be used to compose the following WMS request (link):

```
http://gstore.unm.edu/apps/rgis/datasets/92403ebf-aec5-404b-ae8a-6db41f388737/
services/ogc/wms?VERSION=1.1.1&SERVICE=WMS&REQUEST=GetMap&BBOX=-108,34.7,-106,36.0333&
LAYERS=2007fe_35_county&FORMAT=image/png&TRANSPARENT=TRUE&STYLES=&
SRS=EPSG:4326&WIDTH=600&HEIGHT=400
```

Given that McKinley County NM is contained within the following BBOX: -109.5, 34.5, -106.5, 36.5

Question 1 What is the aspect ratio ($BBOX_{width}/BBOX_{height}$) of this geographic region?

Question 2 What would be the height (in whole pixels - round up) for a map image for this region that is 600 pixels wide?

Question 3 Formulate a WMS request that reflects the values determined in questions 1 and 2 above for the WMS service used above in the examples and for the `g_2007fe_35_county` layer. Include in your answer the complete WMS request that will return a map image.

Question 4 Formulate a WMS request for a 900x600 pixel map image that represents the full 3-degree width of the geographic region, and is based upon the minimum Y value of 34.5 degrees North Latitude. Include in your answer the complete WMS request that will return a map image.
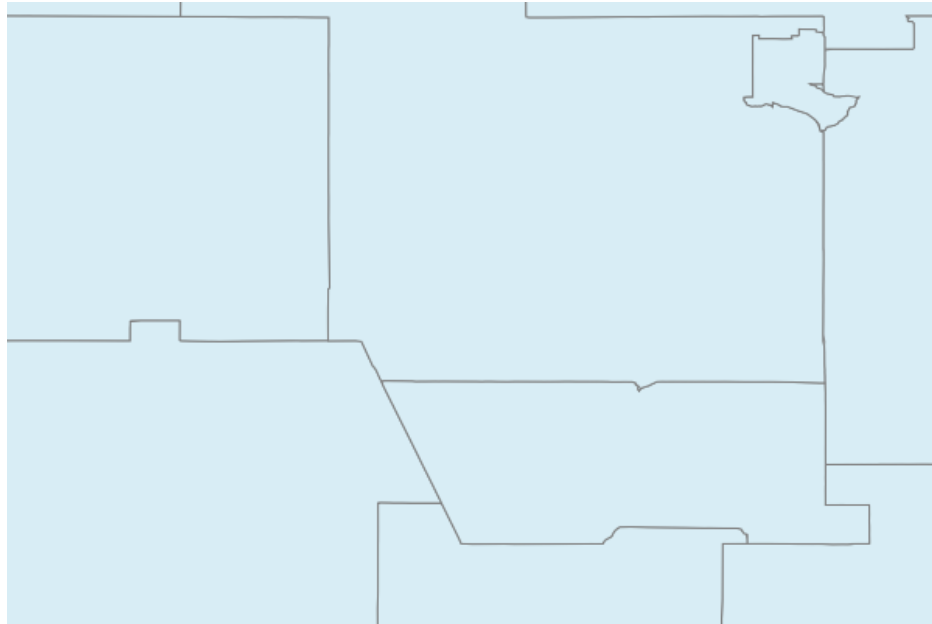
Figure 30.4: Returned map image for the region surrounding Bernalillo County for a WMS request with `BBOX=-108,34.7,-106,36.0333`, `WIDTH=600` and `HEIGHT=400`

Given the following set of `GetMap` requests against the *USGS Topo Base Map - Primary Tile Cache (Tiled)* answer the following questions

1) https://basemap.nationalmap.gov/arcgis/services/USGSTopo/MapServer/WmsServer?request= GetMap&service=WMS&VERSION=1.1.0&SRS=EPSG:4326&LAYERS=0&FORMAT=image/ jpeg&TRANSPARENT=FALSE&STYLES=&WIDTH=1200&HEIGHT=800&BBOX=-107. 1242070,34.7509960,-106.1242070,35.4176960

2) https://basemap.nationalmap.gov/arcgis/services/USGSTopo/MapServer/WmsServer?request= GetMap&service=WMS&VERSION=1.1.0&SRS=EPSG:4326&LAYERS=0&FORMAT=image/ jpeg&TRANSPARENT=FALSE&STYLES=&WIDTH=1200&HEIGHT=800&BBOX=-106. 6867070,35.0426773,-106.5617070,35.1260148

3) https://basemap.nationalmap.gov/arcgis/services/USGSTopo/MapServer/WmsServer?request= GetMap&service=WMS&VERSION=1.1.0&SRS=EPSG:4326&LAYERS=0&FORMAT=image/ jpeg&TRANSPARENT=FALSE&STYLES=&WIDTH=1200&HEIGHT=800&BBOX=-106. 6281133,35.0817417,-106.6203008,35.0869503

4) https://basemap.nationalmap.gov/arcgis/services/USGSTopo/MapServer/WmsServer?request= GetMap&service=WMS&VERSION=1.1.0&SRS=EPSG:4326&LAYERS=0&FORMAT=image/ jpeg&TRANSPARENT=FALSE&STYLES=&WIDTH=1200&HEIGHT=800&BBOX=-106. 6246953,35.0840205,-106.6237187,35.0846715

Question 5 Which layer(s) return map images that display image content (i.e. return a non-blank image)?
Questions 6 What is the difference between these requests? How might this difference influence whether or not a map image with content is being returned?

Chapter 31

Week 8 - Midterm Examination (Take home) & Submission of Portfolio for Review

Week 9 - Spring Break

title: Week 10 - Web-based Mapping Clients - OpenLayers Javascript Framework ...

Chapter 32

Introduction

The Google Maps API provides one method for presenting an interactive mapping tool within a web browser, but there are restrictions for free use based upon Google's license agreement, and the API is completely controlled by Google - changes are limited to those that Google enables. The OpenLayers Javascript framework which (as quoted from the OpenLayers project home page)

> has been developed to further the use of geographic information of all kinds. OpenLayers is completely free, Open Source JavaScript, released under the 2-clause BSD License (also known as the FreeBSD).

Given its Open Source model, OpenLayers is managed as a community software project, with the development of specific capabilities driven by particular project or functionality needs that come out of the community.

This week's class focuses on the basics of designing an OpenLayers interactive mapping client, including

- The required structural (HTML) and behavioral (Javascript) components
- Examples of creating and adding layer objects to a map
- Examples of a variety of base maps that can be added to a map
- The controls that may be added to a map, and the methods for managing and positioning those controls

Expected Outcomes

At the end of this class the students will be able to:

- Create a new web page that includes an interactive OpenLayers mapper
- Define one or more base map layers as part of the mapper
- Define an appropriate map center and zoom level for the desired map
- Enable and position controls within the map

Key Concepts

At the end of this class students will understand that

- OpenLayers is a Javascript framework that enables web-based interactive mapping
- The OpenLayers framework supports the integration of a variety of proprietary and open source base map services
- Map size, center, zoom level, and layers may all be defined through the Javascript API
- A wide variety of map controls (informational and interactive) may be added to maps

Chapter 33

Class Prep

OpenLayers Quick Start Page

OpenLayers Basic Concepts

OpenLayers Workshop

Gratier, T., Spencer, P., & Hazzard, E. (2015). *Openlayers 3 beginner's guide : Get started with openlayers 3 and enhance your web pages by creating and displaying dynamic maps.* Birmingham, England: Packt Publishing. eBook Chapters 1-3. **Read for concepts - the framework has changed enough since the publication of this book that much of the code doesn't work as you would expect**

Chapter 34

Reference Materials

OpenLayers API Reference

OpenLayers Sample Maps

Chapter 35

Weekly Milestone - OpenLayers Mapping

Following the model used in Milestone 3 for your first Google Map web page, you should answer the questions in Learn about what and how you want to map - *relating to a different focus than you have used in your previous assignments*. As you define the type of map you want to build, think about a specific problem or topic that you would like to address with your map.

In this exercise you will be generating the configuration for the base map (i.e. including one or more OpenLayer enabled background layers), adding controls, and defining an appropriate map center and zoom level for the map. You will add your own custom content (i.e. the answers to the following questions) to a free-standing web page that include an interactive mapper and the reasoning behind the design of the map.

Once your have answered the questions in Learn about the map that you wantn to create, refer to the examples in the lecture notes, the OpenLayers Examples (http://openlayers.org/en/latest/examples/), and this week's reading assignment to create a custom OpenLayers map.

Provide the link to your new OpenLayers map in response to the last question in the assignment in Learn.

Chapter 36

Introduction

Background

As we learned in our introduction to OpenLayers in last week's class, it provides a flexible platform for integrating geospatial data from a variety of sources in a single interactive mapping environment. These capabilities include data integration from OGC Web Map Services and a wide variety of data formats for display in the mapper. This week's class provides an overview of the options for the OpenLayers `Map` and `Layer` objects, and specific information about configuring OGC WMS, KML, and Vector Geometry Layers and a brief overview of styling of vector data layers.

Expected Outcomes

At the end of this class, students should be able to:

- Create a new map object
- Add a WMS layer to that map
- Add a KML layer to that map
- Add a variety of Vector Geometry objects to the map
- Define styles for the created KML and Vector Feature objects.

Key Concepts

At the end of this class students will understand that

- There are a wide variety of options that may be provided when creating `Map` and `Layer` objects within OpenLayers
- WMS layers may be added to a map from diverse online servers
- KML files may be added to map, but their potential size can pose a challenge to effective integration into online mapping applications
- A wide variety of geometric objects may be defined and added to a map
- There exist a variety of styling options for vector data that are added to a map

Chapter 37

Class Prep

Gratier, T., Spencer, P., & Hazzard, E. (2015). *Openlayers 3 beginner's guide : Get started with openlayers 3 and enhance your web pages by creating and displaying dynamic maps.* Birmingham, England: Packt Publishing. eBook Chapters 4-6. **Read for concepts - the framework has changed enough since the publication of this book that much of the code doesn't work as you would expect**

OpenLayers 4 Examples:

- *Single Image WMS Example*
- *Tiled WMS Exmaple*
- *Vector Icon Example*

Chapter 38

Reference Materials

OpenLayers API Reference

OpenLayers Sample Maps

Chapter 39

Weekly Milestone - A Customized OpenLayers Mapping Client

Please create a new OpenLayers mapping page that is based upon the initial map (and thematic focus) that you created for last week's milestone, and add the following to your map:

1. Five Vector Features (based upon Point, LineString, or LinearRing Geometries), each assigned its own style.

2. One KML Layer, also styled. Make sure to give some thought to the size of the KML file that you use as large (e.g. > 1-2 MB) KML files can cause slow page loads and in some cases can crash your browser.

3. One WMS Layer.

Submit the link to the live preview of your new OpenLayers mapping client as the response to the Milestone 9 assignment in Learn.

Chapter 40

Deep Dive -

This deep dive is the first of two (plus milestones) that are related to a common theme of taking an online mapping problem from start to finish - from the definition of a problem, through the implementation of services and a basic client interface for the display of data for the specific problem.

Before beginning to work on any mapping problem (online or otherwise), some basic questions need to asked and answered. Please consider and answer (in 2-4 sentences as appropriate for the question) the following questions relating to the problem that you will be working on for the forthcoming assignments:

- What is the high-level description of the problem/question you want to help answer through the presentation of a collection of geographic data over the internet? Please answer in a short paragraph.

- Who is the target audience for the information you want to provide?

- What geographic region does your problem area represent? Please describe it in words (e.g. New Mexico, Alberta Canada, etc.) and define it in terms of a geographic (WGS84) (latitude and longitude) bounding box.

- What types of data do you want to include in your project? Include a description (data content: e.g. elevation data, hydrographic survey, etc.) and types (i.e. raster, vector).

- What projection will you use for the presentation of your project data? Again, describe it, provide your reasoning for selection, and provide the corresponding EPSG code.

- Where do you anticipate acquiring data for the project from?

- What barriers to acquiring and processing the needed data do you anticipate?

While you are going to continue to acquire additional data for the project over the next couple of assignments, begin acquiring data for your selected project now. Specifically, find 5 datasets that are consistent with your description of the problem you are going to work on, and use your desktop GIS and any associated metadata to describe them in brief by answering the following questions:

- What is the name of the dataset?

- What type of data (raster/vector) are in the dataset?

- What is its format?

- What is its coordinate reference system?

- What are the spatial extents of the dataset?

For a dataset to be useful, you should be able to answer all five of these questions about it. Furthermore, you should seek out datasets that are in common formats that you know you can work with in a variety of applications (i.e. QGIS, ArcGIS, gdalinfo, ogrinfo, etc.). If you can't access and use a dataset in one of these applications, you will probably have problems trying to use it in your project.

Also, below are links to the documentation for GeoServer - the platform that we will work with in a couple of weeks for publishing data - relating to the supported raster and vector formats in the default GeoServer installation.

Vector Data: http://docs.geoserver.org/stable/en/user/data/vector/index.html

Raster Data http://docs.geoserver.org/stable/en/user/data/raster/index.html

Chapter 41

Introduction

As we've discussed the components of the client tier of our tiered geospatial services oriented architecture we have concentrated on the open standards that can support client applications and the web-based clients that can consume them. Desktop GIS applications can also consume standards-based services, specifically OGC services. This week's class concentrates on the methods for integrating OGC services into two GIS client applications Quantum GIS and ArcGIS, demonstrating the utility of using external standards-based services as a data and map image source within desktop applications.

Expected Outcomes

At the end of this class, students should be able to:

- Add a WMS service to Quantum GIS
- Add a WFS service to Quantum GIS
- Add WMS, WFS, and WCS services to ArcGIS (if they have access to the required software)

Key Concepts

At the end of this class students will understand that

- The key to configuring a desktop client application is the GetCapabilities request for the needed service
- The GetCapabilities request required by a particular client may consist of a base URL or a complete URL.
- Quantum GIS uses a base URL request model for self-configuration of WMS and WFS services
- ArcGIS uses a base URL request model for self-configuration of WMS, WCS, and WFS services
- Both Quantum GIS and ArcGIS require additional configuration to enable WFS access

Chapter 42

Class Prep

- Quantum GIS documentation, especially
    - Working with OGC Data
- ArcGIS , especially
    - About using OGC service layers
    - Connecting to GIS servers
    - Adding a WFS service to ArcMap

Chapter 43

Weekly Milestone - WMS, WFS and WCS Access in Quantum GIS

While the focus of these instructions is on using QGIS to interact with remote OGC services you may use ArcGIS instead of QGIS if you prefer (and have access to a copy of ArcGIS that has the interoperabilty extension* installed [for the WFS requests]).*

Add three WMS layers to a new map project in QGIS, with at least one coming from each of the following collections of WMS services.

Some things to keep an eye out for:

- Any scale limits described for the various layers
- Layer names can sometimes be a bit confusing
- You can double-check the base URL advertised for the service by reviewing the content of the `GetCapabilities` area of the `service` metadata provided as part of the `GetCapabilities` request. If you can't manually request and review the GetCapabilities XML file for the service, your desktop client may not be able to connect to and retrieve the file for its configuration.

USGS's National Maps Small-Scale Web Services *Page*: http://basemap.nationalmap.gov/arcgis/services/ USGSTopo/MapServer/WMSServer?request=GetCapabilities&service=WMS

NASA Earth Observation System: [https://neowms.sci.gsfc.nasa.gov/wms/wms?version=1.3.0&service=WMS&request=GetC

In your write-up include the names of the layers you added, which service they came from, and screen shots (one for for each of the added layers) showing each of them in the QGIS client interface.

Add three WFS layers to the same QGIS project, with at least two based upon data available from the RGIS data browser (http://rgis.unm.edu/getdata/), and one based on the GeoServer sample WFS service (http:// internetmapping.net:8080/geoserver/ows?service=wfs&version=1.1.0&request=GetCapabilities. In RGIS you can see the available services for a specific data layer by

1. Selecting the collection you want to view by selecting from the directory tree on the left side of the page;

2. Identifying the data sets that have available OGC WMS and/or WFS services as indicated by the "Services" entry for each dataset, where the provided links are for the GetCapabilities requests for the provided services:

Figure 43.1: Illustration highlighting where to see if a spacific dataset has an available OGC service

Important: When adding any WFS layer, you may need to go into the preferences for QGIS and under the "Network" options increase the "Timeout for Network Requests(ms)" value to a larger number than the default 60000 (1 minute). If you don't do this, QGIS might give up on the request before it has been fulfilled by the server. You may also want to zoom into a limited area and check the box in the QGIS "Add WFS Layer …" dialog for "Only Request Features Overlapping the Current View Extent" as this will reduce the number of features recovered - a significant issue if the WFS service is publishing a large number of features.

In your write-up include the names of the layers you added, and the GetCapabilities requests related to those layers. Also include screen shots (again, one for each added layer) showing each layer in your QGIS project.

Chapter 44

Introduction

Thus far we have concentrated on the client side of geospatial services oriented architectures in developing web interfaces based upon the Google Maps API, the OpenLayers javascript framework, and accessing data published using the OGC WMS, WFS, and WCS standards in desktop applications. Starting this week we begin our work on the server side - working with the GeoServer server platform to publish data through the OGC WMS, WFS, and WCS service standards. This work will demonstrate the ease with which you can share data using these standards, facilitating client use such as that that we have seen in our web site and desktop application work.

Expected Outcomes

By the end of this class, students should be able to:

- Place files within the server file system for integration into the GeoServer platform
- Create a GeoServer *Workspace*, *Store*, and *Layer* based upon those data
- Test those layers using the *Layer Preview* tools integrated into GeoServer

Key Concepts

By the end of this class, students should understand:

- The components of a map server platform and their relationship to each other
- The role of a geospatial server within a geospatial services oriented architecture
- The information required about data to successfully configure it for publication within GeoServer
- The stepwise process through which a dataset may be published using GeoServer

Chapter 45

Reference Materials

- Safari Books Online *Fundamentals of Linux: Learn important command-line tools and utilities with real-world examples.* - particularly:
  - Chapter 2: Getting to Know the Command Line
  - 1. Chapter 3: It's All About the Files
- GeoServer Online Documentation: sections Introduction, Getting Started, and Web Administration Interface

Chapter 46

Weekly Milestone - Linux Basics and GeoServer Data Import

Working on the Class Server

For the GeoServer portion of our work, you will be working on a Linux server that has been created for the class. While we won't be doing a lot of Linux work, some basic familiarity with moving around, copying files, and working with files is needed. The class server is running Ubuntu Linux which is a broadly deployed, well supported operating system and computing platform that has excellent support for many Open Source geospatial applications, including those that we will be using in this class.

The first set of exercises relate to learning some basics about working with the Linux Operating system, applicable on just about any Linux server including the class server.

Review (but don't worry about memorizing) the following materials (in addition to watching the video tutorial sections listed above):

Webmonkey "Unix Guide"

Linux Command Line Cheatsheet

QUESTION 1 What command would you use to list the contents of a directory on a linux system?
QUESTION 2 What command would you use to read the "manual page" for a specific command?

Log into the class Linux server - `internetmapping.net:8080/geoserver/web`. *This is different from the address referenced in the below linked videos* The rest of the process is the same as demonstrated in the videos. Your username and password for both the class linux server and the GeoServer have been sent to you via email.

Windows: Open PuTTY on your computer and connect using the SSH protocol (see video demonstration)

Link to the YouTube video demonstration for Windows

Mac: Open the Terminal Application and connect using SSH (see video)

Link to the YouTube video demonstration for Mac OS X

Start a session on the class Linux server, which is located at at the hostname `internetmapping.net` (you will use your class server username and password you received through email to open the connection). **NOTE: the class server is accessed through a non-standard network port. Enter the port number 23 in the connection dialog boxes where there is an option to specify the port. When using the SSH commmand [i.e. on the Mac] include the port number in the connection command.**

For example:

```
ssh -p 23 user001@internetmapping.net
```

After logging in you are in your `home directory` - the directory that is linked to your account on the system, and the directory that you are taken to when you type the `cd` command without any additional arguments.

Adding data to GeoServer

To add data to GeoServer you must have a file location on the server where data files are stored and accessible by the GeoServer.

Task Change into your home directory using the `cd` command without any additional arguments.

Task Copy the folder of sample data files located at `/opt/geoserver/data_dir/general/user000/GeoserverSampleData` by executing the following command from *inside your home directory*.

cp -r ../user000/GeoserverSampleData . (make sure to include final ':')

This will place a copy of folder of data files in your home directory. Rename (using the linux `mv` command) each of the copied files and directories (and their contents) to prepend (and replace mine) your initials at the beginning of each file and directory name. For example, rename `kb_m_3510659_ne_13_1_20110523.tif` as `xy_m_3510659_ne_13_1_20110523.tif`. This will help avoid some issues with layers based on source files with the same name later in our work. **You might find this a faster task using the WinSCP [Windows] or CyberDuck [Mac] utilities instead of the command line**
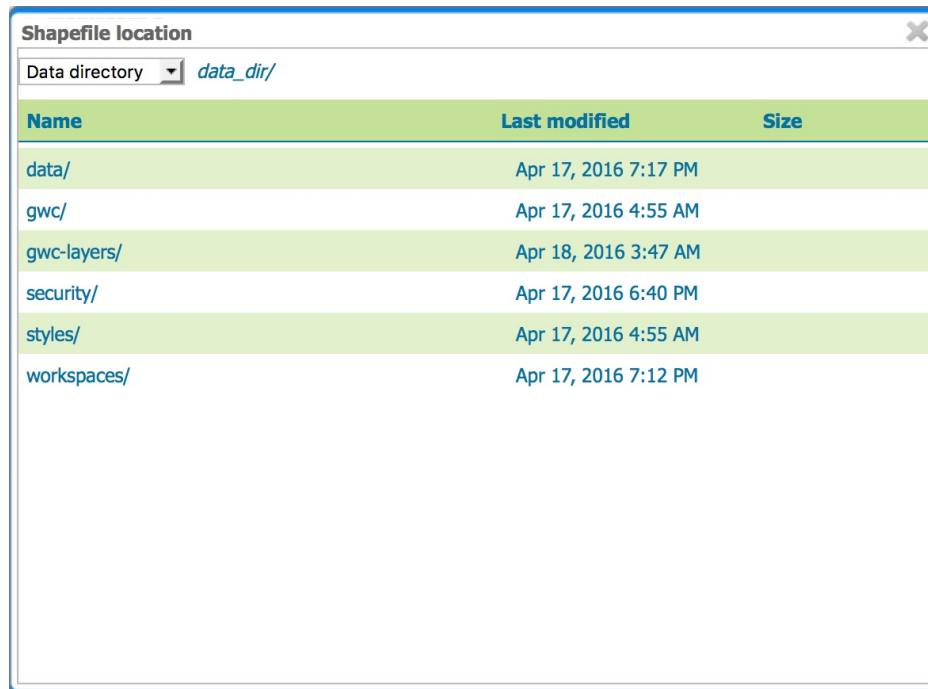
Task Log into the Geoserver on the class server (http://internetmapping.net:8080/geoserver/web/) using the username and password provided by email.

Create a new *workspace* based on your net id. For example `ws_<your netid>`

Create a new *store* for each of the datasets added to your home directory above (**4 .tif files and 3 shape files**). Assign the new store to the workspace that you created above. When specifying the the `Connection Parameters` for pointing to the file you can browse to the location in the server's file system by using the `browse...` link next to the URL field under the `Connection Parameters` section of the store creation page. All of the home directories are in the `general` folder under the `data_dir` in the file browser.

for example

```
file:general/user000/GeoserverSampleData/kb_m_3510659_ne_13_1_20110523.tif
```

Create a new *layer* for each of the *stores* added above. Here are some things to keep in mind:

You may need to designate the SRS for a layer if it can't be read directly from the dataset. Your specify the *designated* SRS using the standard EPSG:XXXX format.

The EPSG code for `GCS_North_American_1983` is EPSG:4269

Question 3 Preview each of your added layers, using the *Layer Preview* tool and the *Open Layers* option to display the data. Include screen grabs of the previews in your write-up.

Chapter 47

Introduction

While integrating data into a web mapping platform is a critical first step, a robust geospatial data service system must also support the development and use of cartographically informative representations of the published data. GeoServer accomplishes this through the integration of the Open Geospatial Consortium's Styled Layer Descriptor standard into the system for publishing alternative styles for vector and raster data products. This standard allows for the configuration of attribute and scale-based representations that may be advertised by name and requested by reference through the published OGC WMS services.

Expected Outcomes

At the end of this class students will be able to:

- Develop a basic SLD for a vector layer within GeoServer
- Define an attribute filter as part of an SLD to allow for differential representation of features based upon the values of one or more attribute values
- Define a scale limit filter as part of an SLD to control the scales at which a defined SLD rule will be executed

Key Concepts

At the end of this class, students will understand:

- The role of the OGC SLD standard in defining and publishing geospatial data via WMS
- The capabilities of the SLD standard to defined representation styles that may be defined in reference to attribute values and WMS request scales
- That logical comparison operators may be used to control the application of style rules to specific features

Chapter 48

Reference Materials

- GeoServer Online Documentation: sections Data Management, Styling

Chapter 49

Weekly Milestone - Styling of Layers in GeoServer

This week's milestone provides an opportunity to experiment with vector layer styling. Please define two custom styles for each of the vector datasets that you added to GeoServer during last week's lab assignment. Take a screenshot of the layer preview for each of your styles - including the options tools above the OpenLayers preview displaying the name of the custom style that is being used for the current map display.

Include in your assignment the layer name, the name of the two custom styles and the associated screenshots for each of the vector datasets.

Chapter 50

Deep Dive - Data Integration and Styling in GeoServer

Add each of the datasets that you acquired for Deep Dive 3 to GeoServer and style at least three of the layers with a custom style designed to best display the data for your envisioned map. Include in your writeup the names of the datasets, associated styles, and screenshots of the layers in the OpenLayers previewer, with the style name displayed in the OpenLayers preview tool set.

Chapter 51

Introduction

Thus far we have discussed the general workflow for publishing geospatial data using GeoServer, the general concepts relating to styling those data for presentation through the web services published by GeoServer, and specific methods for styling vector data, and applying styles to specific subsets of vector data through the use of filters. This week we conclude our discussion of GeoServer with a demonstration of the methods used in the styling of raster data. Without the application of meaningful styles, raster data can be visually unintelligible. You will learn the tools that you have within the OGC Styled Layer Descriptor standard (as implemented within GeoServer) to define and apply meaningful styles to raster data - ultimately producing more useful visualizations of those data.

Expected Outcomes

At the end of the class, students will be able to:

- Define ColorMaps for raster data that produce a continuous gradient of colors, a stepped color ramp between defined raster values, and color assignments for specific raster values.
- Map bands from a multi-band raster file to the RGB colors used to generate a color map image or to a single grayscale band for generating a single-band representation.
- Define opacity for an entire raster dataset or on a per-value level within a ColorMap.
- Apply a variety of contrast enhancements to individual color bands to improve/modify the display of the overall map image.

Key Concepts

At the end of the class, students will understand:

- The concepts of continuous color ramps, color intervals, and colors assigned to individual raster values as alternative representation models for raster data
- The concept of a multi-band raster image and the mapping of specific raster bands into the RGB color space for visual representation
- Two methods for defining opacity for a raster dataset as a whole or for individual values
- The options for and potential effects of contrast enhancement on raster map images.

Chapter 52

Reference Materials

- GeoServer Online Documentation: sections Data Management, Styling

Chapter 53

Weekly Milestone - Create a Final OpenLayers Client

Please create a final OpenLayers mapping client that displays the GerServer-based Styled WMS layers that you created for Deep Dives 3 & 4, focusing on the goals that were laid out in Deep Dive 3. Include in your mapping client a narrative description (a paragraph or two, aimed at a novice user coming to your page for the first time) of the goals and data contained within the client.

Chapter 54

Week 16 - Course Review and Emerging Technologies Discussion

This week is an open class session in which we will discuss any remaining questions relating to the technologies used, next steps for continued use of the developed skills, and specific questions relating to the milestones and deep dive assignments that are due for submission for the second-half portfolio review at the end of the week.

author: Karl Benedict date: GEOG 485L/585L - Spring 2018 include-after: |

...