Developing Mobile Web ArcGIS Applications

Through this practical and accessible guide, you will learn how to build mobile applications using ArcGIS Server from scratch. You will start by covering the differences between mobile web and traditional web development. Factors such as the range of devices, design and layout, functionality, and performance considerations all make mobile development unique.

You will then move on to cover key JavaScript frameworks and libraries for mobile development and cover important structures and concepts including Dojo, responsive design, and the geolocation API, walking through developing your first ArcGIS mobile application in the process. Screenshots, tips, and code are provided clearly throughout to help you through the development process.

Finishing with advanced topics and best practices such as ArcGIS online and PhoneGap, you will finish the book with a complete mobile application and a solid foundation to develop mobile ArcGIS apps.

Who this book is written for

This guide is invaluable to those just starting out with GIS development but will also benefit GIS professionals wishing to expand their development skills to include mobile apps.

What you will learn from this book

- Set up your development environment and understand the key frameworks and tools needed to build mobile ArcGIS apps
- Add advanced tools to your base application including features such as address search
- Provide cross-device support using responsive design using the Bootstrap framework
- Integrate your mobile ArcGIS application with ArcGIS online
- Build high performance, attractive mobile applications using the lightweight, browser-based ArcGIS Server JavaScript API
- Get acquainted with the entire process of creating mobile apps and integrating them with other platforms
- Leverage HTML and JavaScript techniques for mobile development
- Create immersive and engaging views and widgets
- Obtain fine-grained control to provide secure, reliable GIS services

Learn to build your own engaging and immersive geographic applications with ArcGIS
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 ‘Introduction to Mobile Web ArcGIS Development’
- A synopsis of the book’s content
- More information on Developing Mobile Web ArcGIS Applications

About the Author

Matthew Sheehan is the founder and principal of WebMapSolutions (http://www.webmapsolutions.com/). He has over 20 years of experience working with both Esri and open source GIS technologies. Much of his current focus is on helping to expand the understanding and use of GIS. Using web technology, he is working with his technical team to build flexible ArcGIS solutions that can be integrated with existing enterprise systems. Disconnected use of GIS in the browser is a key part of this work.

Matthew has a bachelor's degree in geography from University of London and a master's degree in Applied GIS from the University of Utah.
Developing Mobile Web ArcGIS Applications

Mobile is transforming the GIS industry. Today, the demand from users of mobile devices for maps and location-based information is skyrocketing. ArcGIS is the most widely used GIS platform, with JavaScript as the most popular ArcGIS API. Esri's JavaScript team have been building a rich set of tools for developing ArcGIS Web applications.

This book will teach you how to build web-based mobile applications using the ArcGIS API for JavaScript.

What This Book Covers

Chapter 1, Introduction to Mobile Web ArcGIS Development, covers some of the core elements of developing mobile ArcGIS applications. We discuss how mobile web development is different from traditional web development. These differences include screen size, user interaction, design, functionality, and user and performance considerations. Mobile browsers and different development frameworks are also considered in this chapter.

Chapter 2, Understanding Mobile Frameworks and APIs, discusses some of the basic tools that are used for mobile web development; this includes popular frameworks and design paradigms. Dojo and jQuery mobile are both covered. In addition, simple code examples are included. We finish the chapter with an introduction to responsive, multi-device design and the Geolocation API.

Chapter 3, Building Your First Mobile ArcGIS Application, covers coding using the ArcGIS API. We introduce key tools, including Chrome's DevTools and the local web server installation. Starting with a brief review of basic coding elements, including HTML5, CSS3, and viewport, the bulk of this chapter walks you through code. We move from a very basic ArcGIS mobile application and evolve this code to include an ArcGIS map layer overlay and zooming to a users current location using geolocation.

Chapter 4, Advancing the Basic Mobile ArcGIS Application, is a natural progression from the previous chapter. We advance our mobile ArcGIS knowledge by adding more advanced functionality to our base application. This includes adding popular tools such as feature popups, a legend, finding features, and address search. At the completion of this chapter, you will have progressed from a basic to a more advanced understanding of ArcGIS mobile web development.
Chapter 5, Providing Cross-device Support with Responsive Design, Shows how to build mobile ArcGIS applications which run on all mobile devices. Here we introduce the idea of responsive design using the popular Bootstrap framework. We start with some of the key concepts, then start coding. We not only walk through the development of cross-device ArcGIS web applications with Bootstrap, but also evolve the base application to include a number of popular tools.

Chapter 6, Integration with ArcGIS Online, will focus on ArcGIS Online webmaps. So far, we worked with individual ArcGIS published services. Here, we will build a mobile ArcGIS application that works directly with ArcGIS Online. The application that we evolve in this chapter will include authentication and a webmap list and it will load a user-selected webmap.

Chapter 7, Developing Hybrid ArcGIS Mobile Applications with PhoneGap, demonstrates the flexibility of building mobile applications with the ArcGIS JavaScript API. We can not only build mobile applications that run in browsers, but also convert these applications to hybrid or installable apps that can be distributed to the various mobile app stores. In this chapter, we will provide guidance on working with PhoneGap and Cordova. We will not only go through setup, but also build a basic ArcGIS hybrid mobile application.
We live in interesting times. Mobile and cloud computing are changing how and where we are able to use technology. No longer are we limited to our home or office. Today, technology is accessible and usable at any location and at anytime. By technology, we mean computers, smartphones, tablets, and the new phablets.

Mobility means the ability to change location. This has suddenly made location and location-based data and analysis terribly important. **Geographic Information Systems (GIS)** is a location-focused technology. It provides the means to collect, store, visualize, and analyze any and all location-based data.

Today's cloud-based mobile world heralds the beginnings of a revolution in location technology and GIS.

This book will introduce you to mobile application development using Esri's mobile ArcGIS JavaScript API; it presumes that you are familiar with HTML, JavaScript, and CSS. The API provides a rich set of tools for developing high performance, engaging mobile ArcGIS applications. You will learn about the classes available in the API for mobile development, how to use them in a JavaScript-based mobile web application, and how to incorporate ArcGIS services and ArcGIS Online content to enhance your applications. ArcGIS RESTful web services will also be introduced, which will provide resources via argument/value pairs. The ArcGIS Feature service, that will be introduced early in the book, is one of the many examples. In this chapter, we will go through some of the fundamentals: how mobile development is different from standard web development and design and performance considerations, mobile device types and popular mobile frameworks that are now available, and the more advanced changes that the rapid advancement in mobile- and cloud-based technologies are bringing to the GIS industry.
Introduction to Mobile Web ArcGIS Development

The topics covered in this chapter are as follows:

- Web ArcGIS development
- Differences between mobile development and traditional web development
- Introduction to mobile frameworks
- How mobile web is different from traditional web
- The impact of mobile and cloud-based technologies
- The Cloud on the GIS industry

Fundamentally, Web ArcGIS development is focused on providing users access, through any web browser, to services provided by the ArcGIS platform: ArcGIS Server, Portal for ArcGIS, and ArcGIS Online. Portal for ArcGIS provides the same experience as ArcGIS Online but within an organization's infrastructure (on-premise or in the cloud). This is a particularly good solution where there are concerns around security. The following screenshot illustrates the mobile ArcGIS application running on an iPad:
Often, mobile ArcGIS application begins with an interactive map that consists of a base map overlaid with point, line, or polygon layers. Click or tap on one of these features and a pop should appear listing the feature's attributes.

ArcGIS web applications are not only used for visualization, but they often provide specific functionality too. Tools can be built that give users specific workflows. These tools might include search, identification, buffering, measurement, and more. One of the attractions of ArcGIS technology is the number of RESTful web services that are provided. There is an expanding list of services such as those that transform an address to a location (geocoding), a location's surrounding area demographics (geoenrichment), and projection (geometry). Web services provide the base for the tools that developers build into their mobile web ArcGIS applications. ArcGIS is a service-rich platform that can be easily accessed through one of the many web APIs such as the ArcGIS JavaScript API.

The transition from standard web development to mobile development needs careful planning and consideration. Mobile web development is different. It is generally targeted at a multitude of low-powered mobile computers that are limited in memory, storage, and processing power. Combine this with the range of different mobile device types and screen sizes; design and performance are key considerations when planning the development of any mobile ArcGIS web application.

It is worth discussing some of the differences between standard and mobile web development in a little more detail. There are some key areas that should be mentioned, notably the screen size, user interaction, design, functionality, and performance considerations.

**Downloading the example code**

You can download the example code files for all Packt books you have purchased from your account at http://www.packtpub.com. If you purchased this book elsewhere, you can visit http://www.packtpub.com/support and register to have the files e-mailed directly to you.
Screen size
There is a wide range of mobile device types, from smartphones to phablets and tablets. Screen sizes vary widely across these different devices, ranging from 3.5 inch smartphones to 10.1 inch tablets. Screen size affects how users interact with mobile web applications. Pixel density also varies between devices. For example, map symbology might look good on an older Android tablet but small on an iPad with a retina screen. The following screen shot illustrates different screen sizes:

Interacting with mobile applications
By interaction, we mean how users work with an application, which includes the following:

- Change from mouse to finger interaction, that is, from click to tap
- New mobile data input methods including popup keyboards
- The need to provide user interaction feedback

Next, we will cover these interaction differences in a little more detail.
From clickable to tappable

The following screenshot illustrates the Smartphone ArcGIS finger interactions that we just discussed:

![Smartphone ArcGIS finger interaction](image)

Traditional web development is focused on mouse interaction, where a mouse is a high-precision device. Mobile web development is quite different, as it relies on touch or finger interaction instead. This is very low precision, particularly since we all have different sized fingers. Tapping a button can be problematic and requires that the selectable on-screen items be larger to ensure a good user experience. Map interaction is driven by a finger slide for pan and a pinch to zoom. Zoom sliders are often included in mobile ArcGIS apps and they also provide zoom functionality.
New data input and collection methods

Data input relies on a screen-based, touch-driven keyboard. Usually, multiple keyboards are available; these are for character input, phone or numeric input, and date input, respectively. Voice is another potential source of data input for a mobile device. This provides new and interesting possibilities for mobile web developers.

Mobile device input types

For ArcGIS developers, the built-in GPS that is present on most devices means that current location, or geolocation, is a valuable new data source for use in any mobile application. Similarly, built-in cameras provide both video and still imagery, and these can help to enrich the data used or collected in an application.

Providing interaction feedback

Any interaction should provide users with obvious feedback. When a user taps a button or link, it is good if the item changes state.

Feedback is particularly important to help guide mobile users, when they use your application.
Chapter 1

For example, as shown in the following screenshots, a green-colored button with the label, 'Online', changes to red and the label changes to 'Offline' after a user tap:

![Button changes color and label based on user interaction](image1)

Another good feedback mechanism is a loading animation. This is particularly helpful when loading maps:

![Map loading animation](image2)
Designing for mobile
Careful consideration is needed when designing mobile ArcGIS applications. Key areas to consider include the following:

- Mobile web applications need to be simple and easy to use
- Given the range of mobile screen sizes and pixel densities, layout design becomes important
- Orientation of a mobile device after it is rotated, as this will directly affect your mobile application and the layout of user interface components
- Differences between mobile device brands such as Apple, Google, Windows, and Blackberry
- The target audience for both the application and the device
- Styling differences between traditional and mobile web applications

Again, let's consider each of these design differences in a little more detail.

Simplicity and intuition
Too often mobile web developers try to recreate a traditional web experience which results in an application that is overly complex. Simplicity and intuition are key. Don't try to do all things for all people. Workflows need to be obvious so that if a user selects a menu item or taps a button, then what they see on the screen makes sense. One screen should logically follow the next.

A GIS map is made up of a base map with so-called feature overlays. These overlays are the map layers and represent geographic features through a point, line, or polygon. Points might represent water valves, lines might be buried pipelines, while polygons might represent parks.

Navigation should be simple. For example, if a user has tapped on a point feature and is viewing a full screen attribute inspector, then returning to the map should be easy and obvious.

Focus on designing your mobile web apps so that they are simple and intuitive. Mobile users will stop using a mobile application if it has a cluttered interface and is hard to use.
Layout

Careful consideration needs to go into mobile ArcGIS app layouts. Single column layouts combined with collapsible menus often work best. Traditional ArcGIS web applications are often designed with multiple columns, commonly with a layer list on the left-hand side, the map at the center, and a list of tools on the right-hand side.

The following screenshot illustrates the multiple versus single column layouts that we just discussed:

Collapsible menus provide ways to easily combine simple interfaces with functional options: widgets and tools. Using a responsive design, as we will discuss in Chapter 5, Providing Cross-device Support with Responsive Design, allows a multi-column layout to be adapted into a single-column layout depending on the screen size of the mobile device that is accessing the application.

Orientation

By rotating their mobile device, users can change the orientation of an application to either landscape or portrait. An orientation change affects layout. For example, a map that has a column on the right-hand side might work well in the landscape mode, but it might obscure the map if a user rotates the device to the portrait mode.
The following screenshot illustrates the mobile device orientation change:

![Mobile device orientation change](image)

**Styling**
Finger tap requires interactive items in mobile web applications to be larger than in traditional web applications. Buttons, links, and menu items all need to be larger than their traditional web counterparts. Recommendations on size vary considerably. For example, the recommended size for buttons varies from 44 x 44 pixels by Apple for iPhones to 28 x 28 pixels as per Nokia's smartphone suggestion. Either way, on-screen items on a mobile device need to be easy to select.

We will discuss style sheets or CSS in depth in later chapters. These greatly help developers style their mobile ArcGIS apps. Both Android and Apple provide style guides and these can be found at the following links:

By contrast, some on-screen items are smaller on mobile devices. Popups are a good example of this.

Mobile apps are often styled for a specific device. Apple and Android have styling and design guidelines. These are recommendations for the look and feel of mobile apps on individual platforms. Though it is a little more difficult to style a mobile application for a specific platform using web technology than it is for native mobile applications, much can be done now with style sheets to provide a platform-specific look and feel to your mobile web application by targeting mobile-specific functionality.

Mobile applications are often used for different purposes than traditional web applications. Their requirements and workflows are quite different from desktop browser applications. Thus, functionality is a key consideration. This includes the following:

- Building a mobile application on a mobile device that makes sense from a functional perspective.
- Avoid mirroring, recreating, or reusing a traditional web application for mobile use.
Thinking carefully about your mobile ArcGIS users. Are they GIS or non-GIS users? Functionality will differ depending on the audience. We no longer build web applications that provide tools for everybody (we hope!).

**Mobile targeted functionality**

There is usually a clear divide between what functionality makes sense for mobile use versus desktop use. Questions around current location or use of GPS are usually mobile use cases. More complex, analytical tasks are more likely to be part of the desktop world.

**Focused mobile applications**

When web ArcGIS applications started to become popular, too often developers tried to recreate desktop applications for the Web. This resulted in complex applications loaded with tools that were cluttered and in many ways unusable. Web GIS had a distinctly narrower focus than desktop GIS. As we move to mobile ArcGIS development, developers need to remember lessons that they learned from the early days of the Web. Mobile web GIS again has a narrower focus than traditional web GIS. A good example of a focused mobile ArcGIS app might be one that just provides editing capabilities: add, edit, or delete a point feature. In the mobile world, focused functionality is king.

**Target audience GIS versus non-GIS users**

You need to build your mobile ArcGIS application for your intended audience. If your app is for maintenance workers who need to report on their work of filling pot holes, you will need to build an app that includes all the functional elements—a map with relevant layers, reporting tool, and so on—that are required to complete this task. GIS has traditionally served a GIS-trained audience. As we will discuss later, the combination of mobile and cloud technology has widened the potential user base for applications powered by ArcGIS.

The world of GIS is changing. As the technology becomes more popular, the number of users unfamiliar with GIS is increasing dramatically. Developers are now building seeing it for the first time GIS-powered web applications for a new audience.

Today, a new group of users is accessing and using services provided by ArcGIS. Functionality, design, and workflows need to be carefully considered for this new audience.
Fast responding mobile applications

Though this is not an exhaustive list, the last key difference that we will discuss between traditional and mobile web development is performance.

Users expect mobile applications to be fast and responsive. Think about how people use their desktops or smartphones today. Gaming and social media are extremely popular. High performance is an expected factor. We will be discussing performance and best practices throughout the book.

One area where mobile web is criticized when compared with native mobile development is in the area of performance. Extra effort and attention is needed during mobile web development to optimize performance. This is also true for native apps and backend services.

Network issues potentially plague all mobile applications. Lack of or poor Wi-Fi connectivity can cause any mobile application to be either slow or unusable. Though beyond the scope of this book, advances in offline or disconnected ArcGIS mobile web application are helping to reduce the impact of network issues.

Working with mobile browsers

There are an increasing number of mobile browsers that are now available. Internet Explorer and Safari are the default browsers for Windows and iOS, respectively. Other browsers including Opera and Dolphin are also becoming more popular. Similar to traditional web development, cross-browser compatibility needs to be carefully considered. This is because mobile functionality that works well in one browser might not work quite so well in a different browser. Testing across all popular browsers remains an important part of mobile web development.

Android devices are shipped with a variety of default browsers. These are often described generically as the 'Android browser'. This can present challenges that are described in a presentation that can be found at http://slides.com/html5test/the-android-browser#/.

There are an increasing number of resources such as modernizer, yepnope, and caniuse.com to help you with the challenges of testing.

See the excellent article in Mashable on mobile testing tools. The article can be found at: http://mashable.com/2014/02/26/browser-testing-tools/#.
Web, native, and hybrid mobile applications

There are three main types of mobile applications: web, native, and hybrid. Once, native was the preferred approach for mobile development, but web technology is advancing at a rapid rate and, in many situations, it can be argued to be there is actually a better choice than native. Flexibility is a key benefit of mobile web; one code base is all that is needed for an app to run across different devices and platforms. In the native world, the same mobile application that runs on iOS, Android, and Windows platforms requires three separate development teams. These developers would need to write the code in Objective-C for iOS, in Java for Android, and in .NET for Windows.

In Chapter 7, Developing Hybrid ArcGIS Mobile Applications with PhoneGap we will discuss hybrid apps. These are mobile web applications that can be converted, using technologies such as Cordova/PhoneGap, into installable apps that more closely resemble their native counterparts, again demonstrating the flexibility of mobile web development.

Mobile frameworks, toolkits, and libraries

JavaScript is an implementation of the ECMAScript language open standard. The JavaScript community is worldwide and very active. JavaScript is one of the most widely used languages. Today, there are a plethora of tools that are provided by this community that help JavaScript development. These tools go under different names: toolkits, libraries, modules, and frameworks.

Often the terms library and framework are used interchangeably. There is a subtle but distinct difference between a library and framework called the inversion of control. When a developer's code calls a library, the code is in control. The reverse is true for a framework. Here, the framework calls the developer's code. More simply put, a library is a collection of functionality that you can call, whereas a framework provides automatic flow controls. For all intents and purposes, libraries and toolkits are identical.

In later chapters, both frameworks and libraries will be discussed in more detail. The code examples provided in the code bundle will be built using the Dojo framework, this is the base for the ArcGIS JavaScript API. Dojo helps developers to build dynamic web interfaces.
An alternative to Dojo is jQuery mobile.

There are two other frameworks that are worth mentioning and will form the base for later chapters. These are Bootstrap and PhoneGap.

**Bootstrap**

Bootstrap is the most popular framework for developing responsive mobile applications. This framework provides automatic layout adaptation. That includes adapting to changes in device orientation and screen size. This means your ArcGIS web application will look good and be usable on all mobile devices: smartphones, phablets, and tablets.

The following screenshot shows Bootstrap downloads, illustrating its popularity:
PhoneGap

PhoneGap is a framework that allows developers to convert web mobile applications into installable hybrid apps.

Cordova is an open source framework that is managed by the Apache Foundation. PhoneGap is based on Cordova and PhoneGap is owned and managed by Adobe.

For more information, visit http://cordova.apache.org/.

The names (Cordova and PhoneGap) are often used interchangeably; in fact, the two are very similar, but there is a legal and technical difference.

These hybrid apps can be distributed through the various mobile app stores, just like their native counterparts. In Chapter 7, Developing Hybrid ArcGIS Mobile Applications with PhoneGap we will walk you through the development of a hybrid ArcGIS mobile app. PhoneGap works by providing a web app with a native wrapper around a headless browser. These wrappers are platform-specific and allow your web app to be built to run natively on any of the popular mobile platforms.

The following screenshot illustrates how to convert a web app to hybrid using PhoneGap:
Summary

Mobile ArcGIS web development is an exciting and ever more popular way to build mobile applications. Though the technology is the same as that used for traditional website development, there are many additional areas that developers need to consider when embarking on a mobile web project. These include different screen sizes, user interaction, design, functionality, and user and performance considerations. Flexibility is one of the key advantages of mobile web development. It is now possible to build web applications that can be used on all popular mobile devices and platforms.

In the next chapter, we will consider some of the core building blocks for developing a mobile ArcGIS web application. These are some of the frameworks and APIs that will become very familiar to you and will help you to advance your mobile ArcGIS development skills.
Where to buy this book

You can buy Developing Mobile Web ArcGIS Applications from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.