Using Cloud Services: A Transport Application

Mastering the basics of geolocation is all well and good, and as we learned in Chapter 9, we can have fun working with our coordinates. A huge range of possibilities opens up once we know where we are, and what other services are available to mix our location information with other data. In this chapter, we're going to take our geolocation know-how, and mix in some local searching capabilities, to build a personalized transit application.

Our example transit application will find where you are, and then search for your nearest transit options, be it a bus stop, a subway or train station, or even an airport. With the basics in place, you'll be able to extend our transit application to pretty much any search-plus-location problem.

As a developer, you have many options for how much or how little work you want to do when blending web- or cloud-based services with your own applications. You have a choice of building your own services using publicly available data feeds, using a full service, or building a hybrid. Using a full service would give us nothing to talk about in this chapter, so we have opted for a hybrid approach, where we'll then use our geolocation know-how to use a standard search service to provide useful transit information.

Before we dive into code, examples, and explanations, consider for a minute the various building blocks you probably already use, but may not actively think about. When building any kind of transport or transit application, we want to combine knowledge and data then to present some useful options to the user. These would include:

- **Who am I?** In a way, it can be useful to think of "who" is using your application as a combination of the actual user, and the device they are using along with its capabilities. We'll be reusing the detection code we covered in Chapter 9 to determine if a device supports geolocation.
Where am I? From Chapter 9, we learned how to determine your user's location from the geolocation capabilities of their Android device and their browser. We'll take that fundamental starting point to drive other options.

Where do I want to go? This is always a nuanced question. Does the application user have a location in mind, or a type of transport in mind, or in fact any preference for how to travel between points. The two key points to remember here are not to prematurely limit the options you make available in your application, and that when you do limit options, you do so deliberately and take the user with you. For instance, let them know that other routes or transit methods are available.

What do online services know about me and my location? This is a big area, and covers many generic services. Given a location, and some kind of desired journey or destination, think about what help is available to you as a developer, so that you don’t need to do all the work yourself. For example, APIs for general web searching from companies like Google, Yahoo, and Microsoft can cope with many types of transit search, including finding transit stops, measuring distances, and calculating transit times. Use these so you don’t have to do this yourself (unless, of course, you’re aiming to compete in that space).

What specialized services know about my location? The biggest area to consider, but don’t be daunted. Specialized APIs and services exist for a bewildering array of transit and transport types. A very large number of public transport agencies around the world make available scheduling and routing data, which we'll discuss later in this chapter. However, many also provide a direct API to answer queries for the next service, nearest service, and so on. Aggregators exist to provide pan-provider services. An example is Kayak for airline transit searches. You can take specialized APIs even further, by thinking about (and asking) why your user wants to make a specific trip. If it’s to visit the cinema, eat at a restaurant, or partake in some other event, services like Yelp, TripAdvisor, and others provide APIs to search for specific venues and events. As an application developer, you can create useful applications that combine all of these to target your users.

What can I usefully conclude with this knowledge? Take the data your application gathers about where the user is, where they want to go, and what they want to do when they get there, and you start to picture the flow and the features of your application. Focus on using the building blocks described in the preceding paragraphs, rather than reinventing them, and you’ll build compelling applications that stand out from the crowd. As an example, imagine if you know that users want to travel by boat from Manhattan to Brooklyn to visit a particular restaurant. You could take the experience to the next level, and use the OpenTable online service to check availability, and note that the