Dictionary application for mobile

AT and T Navigator, powered by TeleNav (a GPS provider), has located the desired destinations with a small margin of error and has three ways to enter an address: on the keyboard, online, or by voice activation (say where you want to go - however, on most phones this feature works by placing an outgoing call, which will cost you cell phone minutes). The turn-based directions are spoken and displayed on a 3D map. The service was fairly easy to use, but tracking your desired point of interest (restaurants, gas stations, and even car parks and Wi-Fi hotspots) could take a long time. Like the others, AT-T Navigator provides visual updates to local traffic and gasoline prices; you can even find the cheapest gas within a 15-mile radius. Note: The iPhone 3G ($99) or 3GS ($199 for 16GB or $299 for 32GB, apple.com), available exclusively by AT and T, can locate you not only by global satellite, but also through cell towers and Wi-Fi base stations. It includes a preloaded Maps feature, powered by Google Maps, that allows you to find points of interest around you and get directions, but these are not voice-directed, making them difficult to use while driving. You can download AT and T Navigator for your iPhone, like any other app, at the same price as the owners of other GPS-enabled AT and T phones. att.com/navigator. Review: September 2009Price When reviewed: $9.99 per month or $2.99/day This content is created and maintained by a third party, and imported on this page to help users provide their email addresses. You may be able to find more information about it and content similar to piano.io A dictionary is a reference book or an online resource containing an alphabetical list of words, with the information given for each word. Etymology: From Latin, to say S.I. HayakawaThe writing of a dictionary . . . is not a task of putting in place authoritative statements about the true meanings of words, but a task of recording, to the best of its ability, what various words have meant to authors in the distant or immediate past. The author of a dictionary is a historian, not a forensic pathologist. If, for example, we had written a dictionary in 1890, or even until 1919, we could have said that the word broadcast means broadcast (seed, for example), but we could not have decreed that from 1921, the most common meaning of the word should become broadcasting audible messages, etc., by radio transmission. To regard the dictionary as an authority is therefore to credit the author of the dictionary with gifts of prophecy that neither he nor anyone else Has. By choosing our words when we speak or write, we can be guided by the historical record that the dictionary offers us, but we cannot be related to it. Looking under a hood, we would normally have found, five hundred years ago, a monk; today we find a motor engine. Stephen Frye, A dictionary is an observatory, not a conservatory.R.L. Trask[T]he colloquial term that an English word of English only if it is in the dictionary is wrong. A word exists if people use it. But this word may not appear in a particular dictionary published at a given time because it is too new, or too specialized, or too localized, or too confined to a particular social group to make this edition of the dictionary. Thomas JeffersonDictionaries are only the custodians of words already edited by usage. The company is the workshop in which new ones are developed. David WolmanMany people mistakenly credit [Samuel] Johnson for writing the first English dictionary. This achievement belongs to a man named Cawdrey, who, 150 years before Johnson, published A Table Aphabetical. It was only 144 pages long and defined some 2,500 dificult words; the rest of the people were supposed to know. With its focus on increasing vocabulary, Cawdrey's book is a bit like modern titles that help you pump your aresenal speech before attacking the SAT or making war in the corporate world. Steven PinkerAlthough dictionaries are powerless to prevent language conventions from changing, this does not mean . . . they cannot spell out the conventions in force at any given time. That's the raison d'être of the American Heritage Dictionary's usage panel - which I chair - a list of 200 authors, journalists, publishers, academics and other public figures whose writing shows that they choose their words carefully. Each year, they complete questionnaires on pronunciation, meaning and use, and the Dictionary reports on the results in usage notes attached to entries for problematic words, including changes in repeated newsletters over the decades. The user panel is intended to represent the virtual community for which careful writers write, and when it comes to best practices, there can be no higher authority than that community. Keith Denning[E]ven the greatest dictionaries cannot capture all possible words in the language. The number of possible word combinations of word elements like pre-, pter, and the countless scope and quantity of speaking and writing made in English require that dictionary publishers limit themselves to listing only the most common words in a language, and even then, only those used over a considerable period of time. Dictionaries are therefore always at least slightly obsolete and inaccurate in their descriptions of the language's word stock. In addition, the use of many words is limited to specific domains. For example, medical terminology implies a huge number of words unknown to those who are not part of the medical community. Many of these terms never enter the general dictionaries of the language and do not only found in specialized medical dictionaries. David Skinner's recent relationship with lexicography has left me certain of a couple of things. The first is that no dictionary contains all the words of the language. Even an unaddebred dictionary is, well, abbreviated. Science, medicine and technology generate word gobs that never make them Dictionary; many foreign words that appear in English-language contexts are left out. Many words are invented all the time, whether for commercial reasons or to amuse his friends or to insult his enemies, and then they simply disappear from the recording. Another is that dictionary users and dictionary manufacturers sometimes have very different notions of what a dictionary is for. One might think that it is a legal code for the language; the other considers it to be a very partial report. We want clear answers on spelling and meaning and grammar and use; the other is about neutrality, and the more serious it is, the more the person is wary of imposing his own notions of good English on the language itself. R.L.G.Macmillan, a publishing house, has announced that it will no longer print dictionaries. And yet he announced it in a tone not of sadness, but of excitement: Coming out of print is a moment of liberation, because finally our dictionaries have found their ideal medium. Michael Rundell, the editor-in-chief, makes a compelling case. Updating the print edition takes five years, while new words are constantly entering the language, and existing words find new meanings. Space constraints limit the actual value of the dictionary. And the points in favour of electronic dictionaries are even more convincing than those against printed dictionaries. Hyperlinks allow you to quickly learn more about the related elements. Audio pronunciations beat transcripts in obscure formats. Photos and even videos are a nod to include. Blogs and other meta-contents enrich the experience. Electronic data storage has already revolutionized lexicography. Huge searchable text corporas allow dictionary manufacturers to find words and uses that are older and rarer than ever before. Having vast, rich and growing data in the dictionary, and a linked and static product that comes out, seems absurd. Dave BerryIf you have a pretty big dictionary, pretty much everything is a word. Ogden NashSeated one day to the dictionary, I was quite tired and also quite uncomfortable, Because a word I had always liked turned out not to be a word at all, and suddenly I found myself among the v. And suddenly, among the v, I came across a new word that was a word called velleity, So the new word I found was better than the old word I lost, for which I thank my tuténaire deity . . . Pronunciation: DIK-shun-air-ee When you develop an application, you choose one of two paths: hybrid or native. Hybrid apps use Apache Cordova, which runs your app in a full-screen web browser phone, but allows access to the platform to the basic functions of the device. The obvious advantage for web developers is that their existing HTML5, CSS3 and JavaScript skills are immediately transferable. It is also possible to take advantage of frameworks such as jQuery Mobile or KendoUI Mobile, which deal with tedious routing and styling. The downside? You have to be careful about optimization, because handling DOMs can be expensive on a mobile mobile Native applications, on the other hand, take advantage of frameworks such as Xamarin or NativeScript, which abstract some of the basic native tools to a common layer you build with. They then compile up to an app that can run on the device's platform (iOS, Android and so on). Therefore, with native applications, you can use pure native UI elements accelerated by the hardware on the device. This approach also generally means that you have full platform-level access to APIs, so you're not limited to available Cordova plugins. Hybrid versus nativeIt can be difficult to tell if a mobile application is hybrid or native. A well-written hybrid application should not look or behave differently from its native equivalent, but they often do. A hybrid app needs to manipulate the DOM (which is an expensive task), and any noticeable delay can sour a user on your app and lead to bad reviews in the App Store. LinkedIn and Facebook have both left HTML5 behind in favor of native apps, and they both have a pretty significant developer talent behind them. What if you could use your JavaScript skills to write a native mobile application, using much less code than a hybrid mobile app, in a fraction of the time, and have all this data exposed to your frontend website users at the same time? With Telerik NativeScript and Sitefinity CMS, all this is possible. The introduction of NativeScriptNativeScript is a framework that allows developers to use pure JavaScript to create native multiplatform mobile applications for Android, iOS and Windows phones, using much of the same code (although you also have the ability to write platform-specific code, if necessary). JavaScript is the underlying language, while the UI layouts are described by XML. NativeScript then converts the XML layout into native UI components of the platform, so there is no compromise in the user experience. You have access to all the power, flexibility and scalability of JavaScript to build your business logic, and that business logic will remain the same, regardless of the targeted platform. NativeScript Run Time provides full access to the underlying native APIs, such as the camera, location and file system, via a JavaScript abstraction layer. iOS, Android and Windows all provide JavaScript virtual machines that allow users to access their native core. Trying your code with NativeScript is as simple as holding three fingers on the companion app after you have XML, JavaScript and CSS. This propagates changes almost instantly - no reconstruction or deployment required. Enter SitefinitySitefinity is a content management and customer analytics platform for Telerik. With the license, you have access to a suite of HTML5 controls in the Kendo UI platform. Sitefinity comes with a mobile module that lets you expose your data in a Kendo UI Cordova mobile hybrid app, and generates an entire frame through which you can view and edit your data with just a few clicks. The Kendo-Cordova code required to perform tasks such as is not insignificant. However, with a NativeScript application, we can extract much of the basic plumbing and simply focus on the data. Surprisingly little code is needed - and on the Sitefinity side, no code at all. Of course, at the moment there is no one-click button that will allow you to generate a NativeScript application again in Sitefinity. However, you can visit the Telerik platform now and use its ScreenBuilder tool to do the same effectively. It will basically let you design your app screens, you just need to plug in your data. I recently developed a native application using NativeScript and Sitefinity and was blown away by the ease with which it was. In this tutorial, I'll help you through, step by step, how to build an application with NativeScript and Sitefinity, and demonstrate the elegance and simplicity of the approach. Modules and TasksIn the sitefinity backend administrative user interface, start by defining your new Sitefinity module. Create a type of content (e.g., task) and designate the type of parent content (if any). Then add the appropriate fields and set them using the drop-down menu. Click 'Finish' and Sitefinity will generate all the backend database tables for you. Sitefinity's module builder quickly facilitates the creation of complex dynamic objects that you can use to store your data Now you can start creating data. The Sitefinity interface makes it easy to create tasks for your module. Just click the 'Create a Task' button on the toolbar and the user interface will allow you to enter new items with the fields you've just set. Sync OptionsThe next step is to sync your data. For this, I recommend using Telerik Backend Services. This is a simple, fast, NoSQL database, and no additional processing is required to get the data. If your site leaves, mobile app users can still browse and edit data, and when it lights up, the system triggers sync at the defined interval, and then consumes all the changes. Sitefinity CMS leverages ServiceStack technology to provide and support some of its features. However, if you want to use ServiceStack to develop and customize the CMS, you guess buy the respective ServiceStack licenses. Data SyncTo synchronize the data you've entered in the Sitefinity platform, first select your new module by checking the box provided, then set the sync interval. Whenever interval travel, reconcile two-way changes with telerik platform backend services. This means that if you add new data to the Telerik Platform's Backend UI, it will automatically sync to Sitefinity at the next sync interval. Once you save the connector, a sync button becomes available, allowing you to bypass the wait time and sync the data manually. By clicking on the Details link at the top right (under Bandwidth) will send you directly to the Telerik platform and your new backend service project. To add your project to the Telerik platform, go to My Workspace to set up the services. Click 'Add to the Project' in 'Cloud Data Services' box. Finally, select Types from the menu on the left. You need to see all the tasks you've created in the CMS appear. Create the NativeScriptTelerik Platform project works with the concept of workspaces, and each workspace can contain one or more types of projects. The data project set for this tutorial is the backend data storage part. Now we need to add a new type of project. Start inside your root taskspace and click the 'Create a Project' button. Now choose 'AppBuilder Native Project'. Once selected, you can choose to create a blank project, load a pre-built sample, or clone a repository. For this tutorial, I chose the example 'NativeScript MasterDetail', because it is a good starting point from which to consume a list of data. Then we will remove the code that populates the list and replace it with the Telerik backend code. We will also change the script of the detail pair to allow data editing/recording to the backend. It's easy to add or delete services from your Telerik platform project To edit code and create apps, you need to use the Telerik AppBuilder. This comes in several flavors: control line interface (for use with sublime text, for example), Windows client, visual studio extension, or even a browser variant! Personally, I prefer the Windows Client AppBuilder, which allows me to pull down the same project on multiple PCs without additional configuration. Check out the GitHub repo accompanying this article to see how little code is needed to list tasks on the main list view, and create the native app that runs on iOS, Android and Windows. In addition, there is no necessary coding in the Sitefinity platform. Testing the appThe companion nativescript app is a test utility for those who use Telerik Platform and Telerik AppBuilder to develop native mobile applications with NativeScript. After installing the companion nativescript app, you can use AppBuilder LiveSync to instantly view and test code changes on your device. Just hold three fingers on our device screen for a few seconds to pull the tasks of the Telerik Backend.Click a task to get into edit mode and then hit save to push the description and completion state back to the Telerik Backend using the pure JavaScript SDK. If you check the code, you'll notice that you can even use JSON.stringify to display debugging data in a dialogue. The end result that's all! As you've seen, with NativeScript and Sitefinity, you can create a fully functional native mobile with very little code. The result will be a beautifully rendered mobile app that provides a native user experience. All this while using your existing JavaScript skills. Words: Steve McNiven-ScottSteve is a Telerik/Sitefinity e-fan-gelist (it's like an evangelist, but you don't get paid). This article originally appeared in issue 271 of net magazine. Did you like it? Read this! These! These!