developerWorks®

No programming? No problem! Build a mobile app using drag-and-drop

Try the IBM RapidApps development environment on the Bluemix cloud platform

Kevin J. Gilhooly (https://www.ibm.com/developerworks/community/profiles/html/profileView.do?userid=0100003TC7&tabid=dwAboutMe)
IBM Consulting IT Specialist
IBM

29 September 2014

Create a small application to help users remember where they parked their car. Because the RapidApps service in IBM Bluemix helps you build mobile-friendly applications, it's well suited to the task of taking a quick note on your mobile phone of exactly where you car is parked. Once you create the app, you can deploy it to the cloud, so anyone can use it.

If you need an application development environment for building and deploying (relatively) simple **CRUD** applications, consider RapidApps. The RapidApps environment gives you easy drag-and-drop tools to develop applications that are primarily designed to process data stored in logical tables. The development environment is designed for programmers and non-programmers alike.

This tutorial shows you how to create a small application to help you remember where you parked your car. Because RapidApps builds mobile-friendly applications, it's well suited to the task of taking a quick note on your mobile phone while you are in the parking lot.

“ I could never remember where I parked, so I decided to write a cloud-based solution I could use from my phone. ”

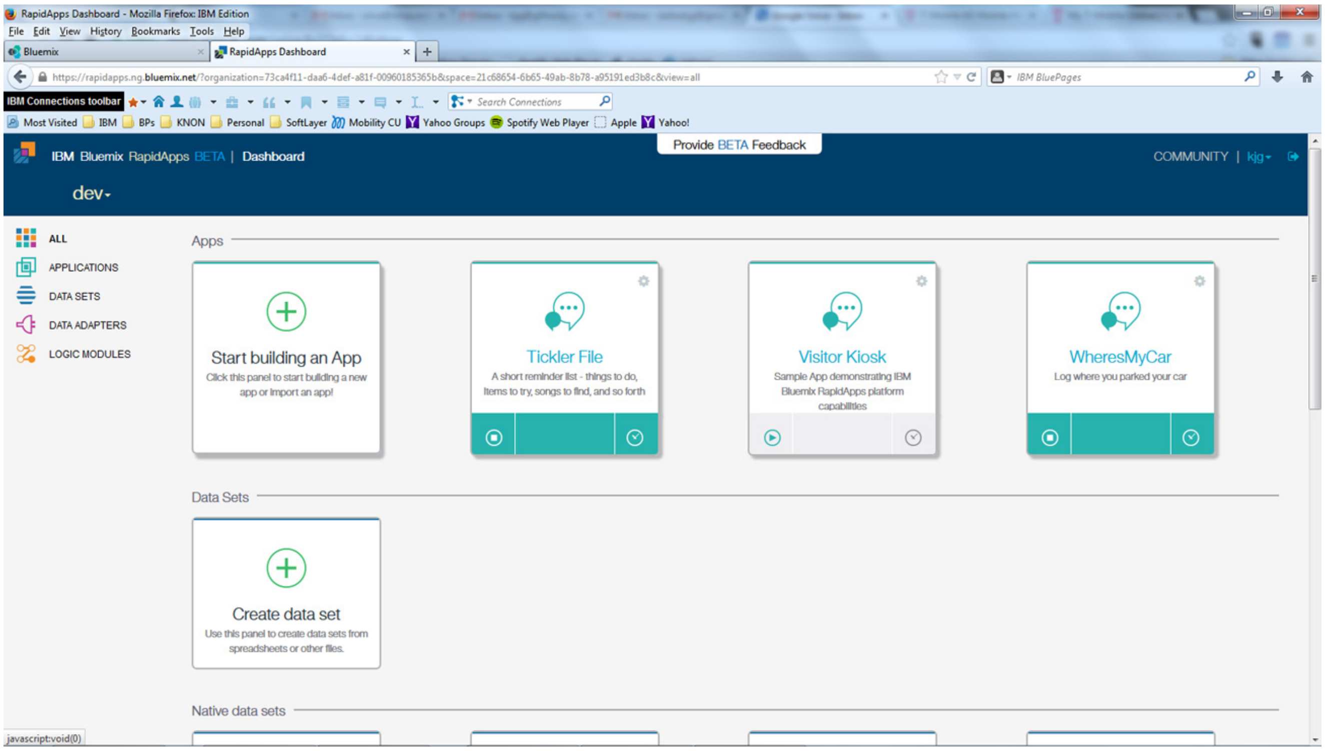
Run the app

What you'll need to build a similar app

- A Bluemix account, so you can deploy your completed app. RapidApps is built on the IBM Bluemix environment.

Step 1. Display your dashboard

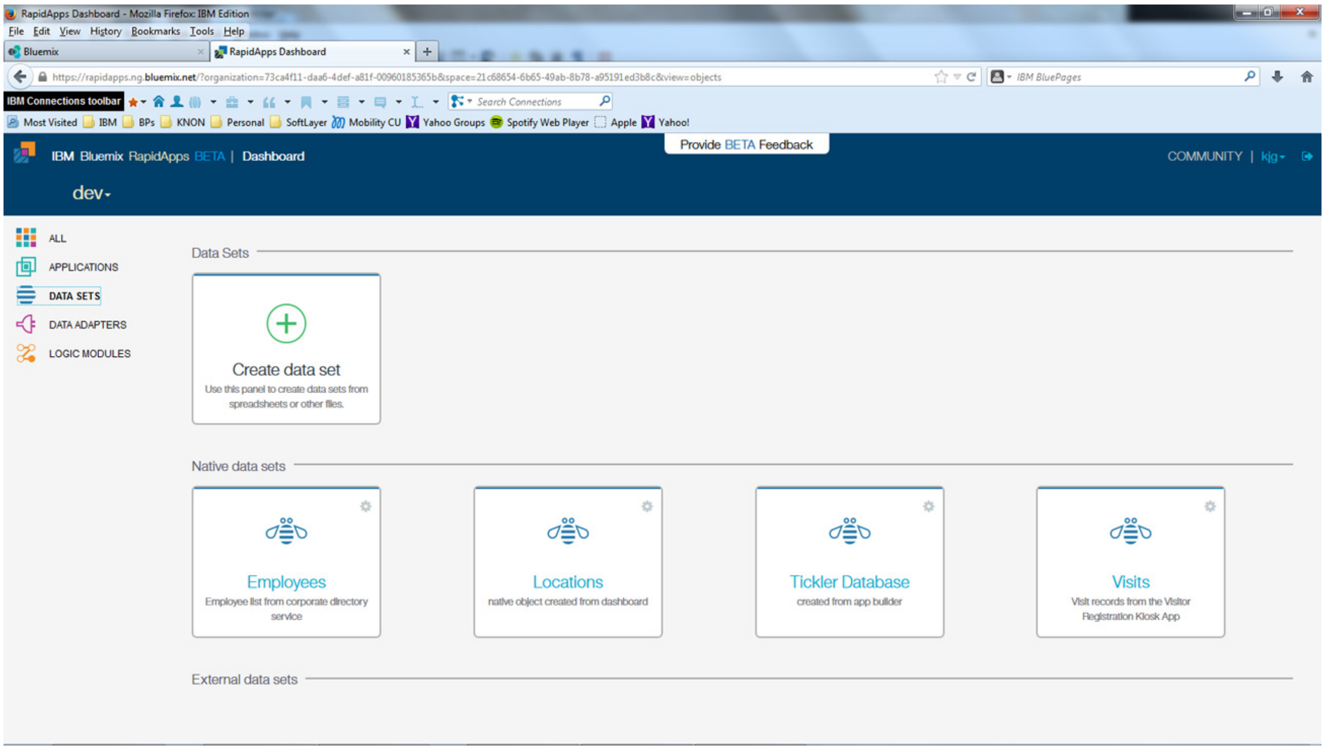
1. Go to the RapidApps Community on developerWorks.
2. At the top right corner, register or sign in using your IBM ID.
3. Open a dashboard by clicking **Open your RapidApps dashboard**. The dashboard displays all of your applications, data sources, and code. When you first begin, you may have a demo application already defined: the Visitor Kiosk. (This sample application shows how to allow visitors to an office to sign in and notify their contact they have arrived.)


IBM Bluemix RapidApps BETA | Dashboard
Provide BETA Feedback
COMMUNITY | kjg
dev
ALL
APPLICATIONS
DATA SETS
DATA ADAPTERS
LOGIC MODULES
Apps
Start building an App
Click this panel to start building a new app or import an app!
Tickler File
A short reminder list - things to do, items to try, songs to find, and so forth
Visitor Kiosk
Sample App demonstrating IBM Bluemix RapidApps platform capabilities
WheresMyCar
Log where you parked your car
Data Sets
Create data set
Use this panel to create data sets from spreadsheets or other files.
Native data sets

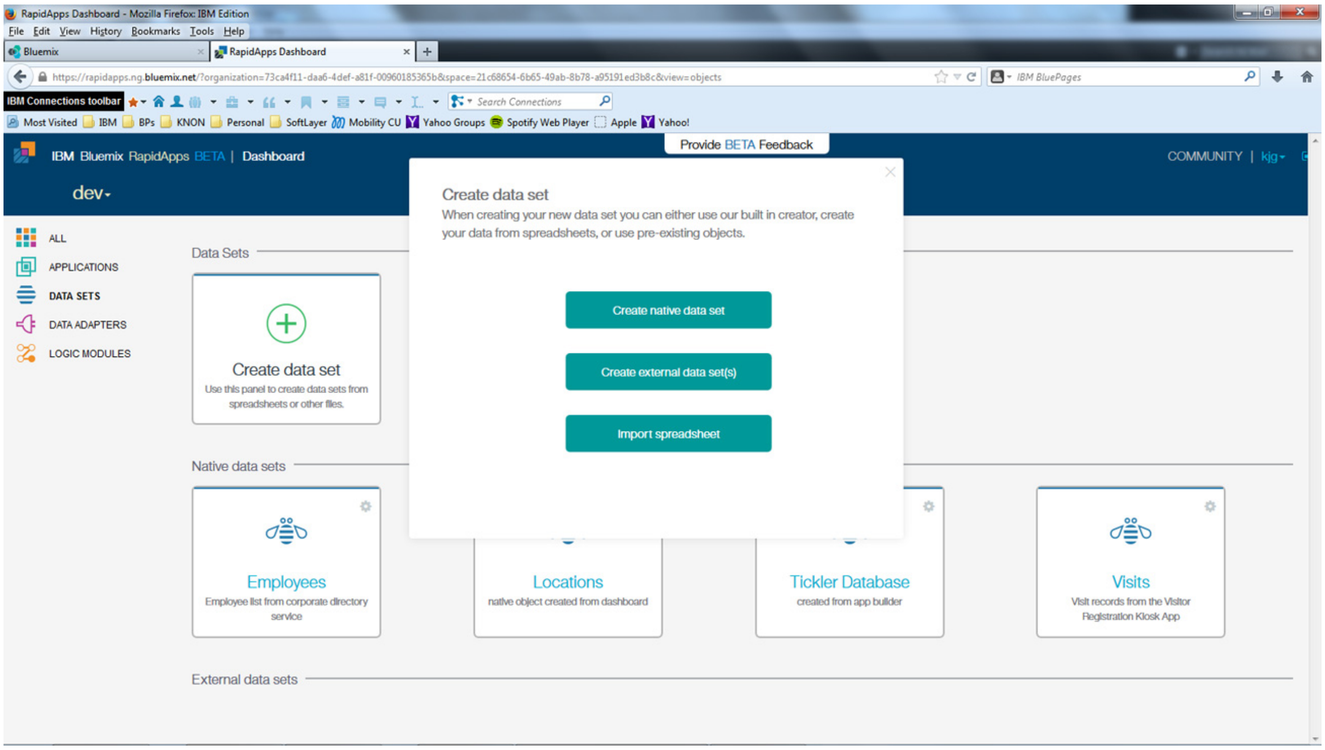
Step 2. Define data

1. To start, you will create the table that will hold your data. In RapidApps, this is called a data set. Click **DATA SETS** in the left menu to show all the data sets that have been defined. You should see the data sets used by the demo kiosk

application. You can ignore them for now. You will create your own data set.


IBM Bluemix RapidApps BETA | Dashboard
dev
ALL
APPLICATIONS
DATA SETS
DATA ADAPTERS
LOGIC MODULES
Data Sets
Create data set
Use this panel to create data sets from spreadsheets or other files.
Native data sets
Employees
Employee list from corporate directory service
Locations
native object created from dashboard
Tickler Database
created from app builder
Visits
Visit records from the Visitor Registration Kiosk App
External data sets

2. You need to create a **Places** data set, the table where the parking locations are stored. You can create a native data set, use existing data, or import a spreadsheet. For this application, you will create a native data set and define the data you want to store. Click **Create data set** and then choose **Create native data set** from the pop-up menu.


Create data set
When creating your new data set you can either use our built in creator, create your data from spreadsheets, or use pre-existing objects.
Create native data set
Create external data set(s)
Import spreadsheet

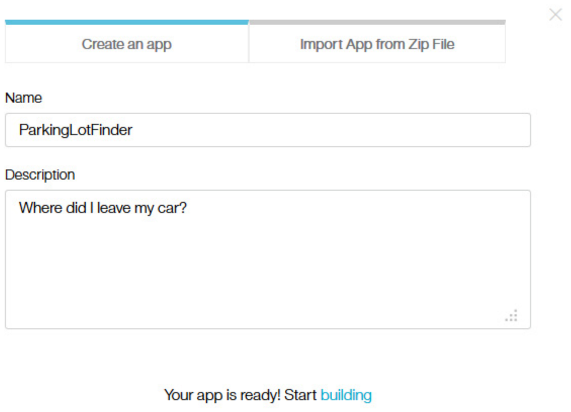
3. Name the data set `Places`. Click **Create**.
4. You will need to define the data set's attributes. There is room for the first attribute in the table by default. You can press **New Attribute** to add additional attributes. (If you make a mistake, press the trashcan icon beside an attribute to remove it.) Add the following attributes:

Driver	Text
Lot	Text
Level	Text
Space	Text
Date	Text
Notes	Text

5. When you have finished, click **Create** to create your data set. You have now added a data set to your RapidApps environment. This data set can be used in other applications that you build. The new table will appear with any other data sets already defined.

Step 3. Create an application

1. Now it's time to create an application to store, update, and display your data. Select **APPLICATIONS** from the left menu. Press the **Start building an App** button to add your new application.
2. Name your application and fill in a description so you can remember what it does. Click **Create** to generate the application.



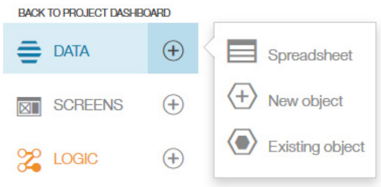
Name
ParkingLotFinder
Description
Where did I leave my car?
Your app is ready! Start building

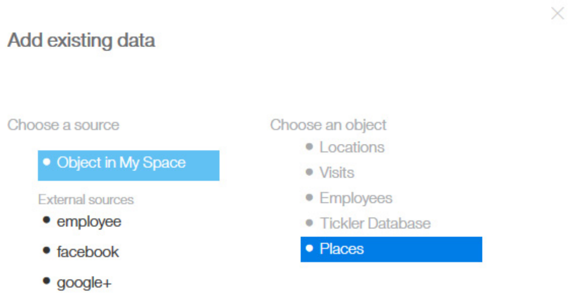
3. The dialog will display that your application is created. Click the **Start building** link to begin building your application.

Step 4. Define application data

Your application opens in a new browser window. Applications can have three components: Data, Screens, and Logic. For basic applications, you may not need any custom logic, but the option provides more flexibility in more complex applications.

Screens are the displays for your data. Data can be defined in the application or can already be defined. You defined your data already, so you will just use it in your application.

1. Select the **DATA** button by clicking the plus sign to its right. Select **Existing object**.


2. Select **Object in My Space**. This lists all objects that have been defined already. You should see the object you just created plus any objects from the demo application. Choose **Places**

Add existing data

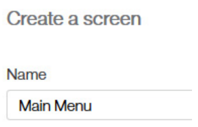
Choose a source
- Object in My Space

External sources
- employee
- facebook
- google+

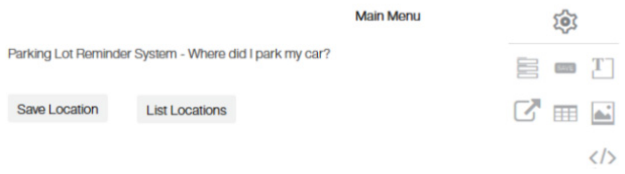
Choose an object
- Locations
- Visits
- Employees
- Tickler Database
- Places

and click **Next**.
3. You are prompted to create default screens for the data. You are going to create your own screens, so you do not need the default ones. Leave both prompts unchecked and press **Next**. You now have a data object associated with your application. You will see it listed under Data on the leftmost side of the screen.

Step 5. Define screens

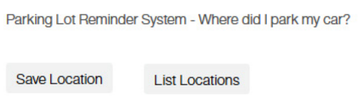
1. Click the plus sign next to Screens to begin setting up the user interface for your application. This will add a screen, which will be the opening screen of the application. Name the screen

Create a screen

Name: Main Menu

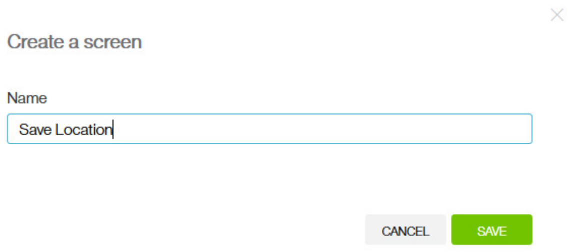
Main Menu.
2. You now see a blank palette with design elements on the right side. You can add descriptive text, buttons, links, and data. You will have only buttons on the Main Menu, since it will let you navigate to other parts of the application.

Main Menu

Parking Lot Reminder System - Where did I park my car?

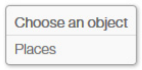
3. Drag a text object onto the screen. Because this application will be used on mobile devices, you probably want the screen to start at the upper left corner. Double-click the text to edit it and change the text to describe your application, such as `Parking Lot Reminder System - Where did I park my car?` Click outside the object to save it.
4. Drag a button onto the screen. Change the text to `Save Location`. Add another button called `List Locations`. You will set these up to point to your other screens later on.



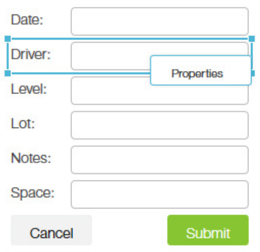
5. Now, you will create the input screen for the location where the user's car is parked. Press the plus sign beside Screens again, and name the screen `Save Location`.



6. Drag a form onto the screen from the design elements. Click **Places** to choose it as the collection for the form. You have now linked your data set to an input screen. This will be the screen the user uses to record the car's location.



7. Click the **Driver** line in the table to select it. Right-click the line to see the pop-up menu. Select **Properties**. (If you see properties for the collection, you didn't select the line first!)

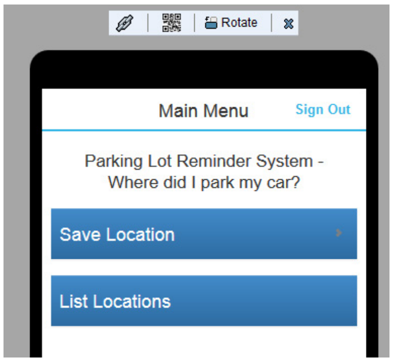


8. In the Value field, enter **USER_ID**. This way, if a user is logged in, he or she won't have to enter a name. You want a name field so multiple users can share the application. Click **Save**.
9. Select the Main Menu screen from the list of screens. Right-click the **Save Location** button and select **Properties**. Set **On click, navigate to** to Save Location (the screen you just created.)

Save the changes. Now, when the button is clicked, it will change to the Save Location screen. You will test that next.

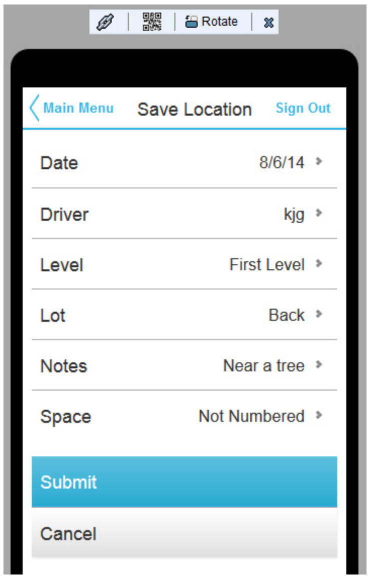
Step 6. Test the basic application

1. On the right side of the screen, click the eyeball icon to see a preview.



Parking Lot Reminder System - Where did I park my car?

2. Notice that in the Mobile Browser Simulator menu, you can add multiple devices to see what your application will look like on various form factors. You can also rotate the view to see what it will look like if the device is rotated. Click **Save Location**.
3. The input form for the Places table should appear. Here you can record the location of your car. Notice that the Driver field is pre-filled, since you set it to be the userid. (This will work as long as you are logged in. Since we are in the development environment, we are logged in to IBM Bluemix.) Add values for each of the other fields. When you're finished, press **Submit**.



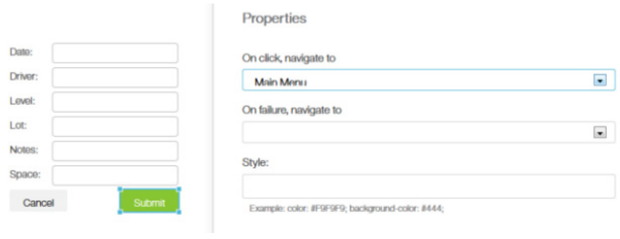
Date	8/6/14
Driver	kjg
Level	First Level
Lot	Back
Notes	Near a tree
Space	Not Numbered

4. A check mark appears and then disappears. The screen returns, which is a function you probably do not want. You should have the screen return to the Main Menu after saving a location. You can repair that easily in the next step. Close the Mobile Browser Simulator window (or browser tab). Return to the development environment.

Step 7. Add functionality

1. First, let's fix the problem in the Save Location screen. Select it from the list of screens. Click the **Submit** button. Right-click to select the **Properties** menu. Set **On click, navigate to**

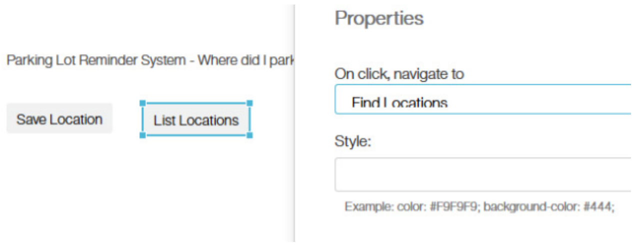
property to Main Menu. This will ensure that, after a location has been entered, the Main Menu will return.



On click, navigate to: Main Menu
On failure, navigate to:
Style:
Example: color: #F9F9F9; background-color: #444;

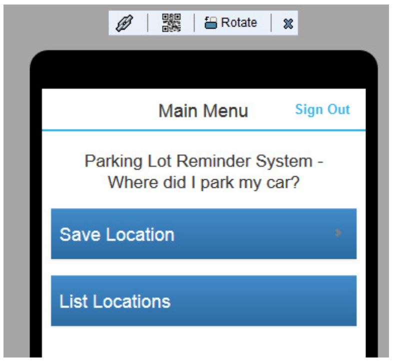
Note that you can also set an error screen. Since this is a demo, we assume everything will work perfectly. If you like, you can test the new functionality by viewing the preview again and adding another parking place record.

2. Now you have a way to store the locations, but you can't find them again! You need to add another screen. In the development environment, click the plus sign beside Screens. Enter **Find Car** into the Name field and click **Save**.
3. Drag a table from the design elements on the right to the screen. Click **Places** to select it as the collection.
4. Switch to the Main Menu screen. On the **List Locations** button, set the **On click, navigate to** property to **Find Car**. Click **Save**.

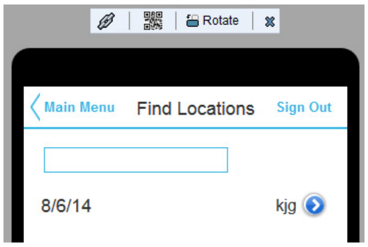


On click, navigate to: Find Locations
Style:
Example: color: #F9F9F9; background-color: #444;

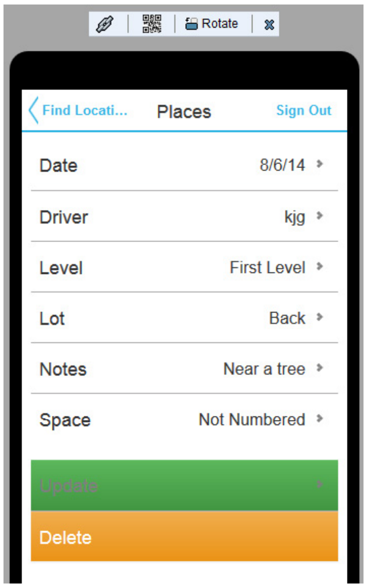
5. Preview your application from the Main Menu.



6. Click **List Locations** to make sure that you see the table listing all of the locations that have been saved in the database.



8/6/14 kjg

7. You should see at least one record. Click the arrow next to the user name to see the entire record. 

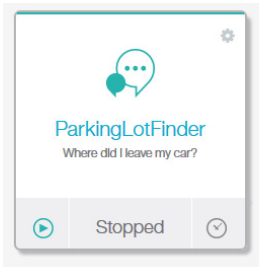
Date 8/6/14
Driver kjg
Level First Level
Lot Back
Notes Near a tree
Space Not Numbered

   Note that **Update** and **Delete** buttons have been added to the form automatically.
8. You can click **Find Car** (at the top left of the preview) to return to the previous menu. This is the standard navigation for RapidApps. (The Find Car screen also has a Main Menu button because we added it.)

You have now created a RapidApps application to store your car's location in a parking lot. Congratulations!

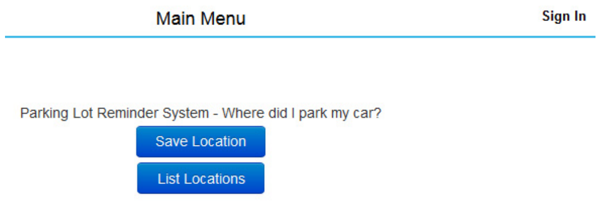
Step 8. Publish your application

Now that the application is completed, you can publish it to the web, so that many users can save their parking locations.

1. Return to the RapidApps dashboard by clicking the link above the Data button on the left side of the screen. (You could also close the tab for your application's development. You should still have the RapidApps dashboard open.)
2. Your new application is now listed on your dashboard. Note that it is stopped. It is not running as a production application yet. Press the **Run** button on the bottom left of the application's icon. 

3. You are prompted for a hostname. This is the name that will be used in IBM Bluemix to identify your application. It must be unique within IBM Bluemix. For that reason, it might be a good idea to use your initials, name, or company name as part of the hostname. For this demo, I chose `ParkingLotFinder`. Enter your own hostname and press **Start**.

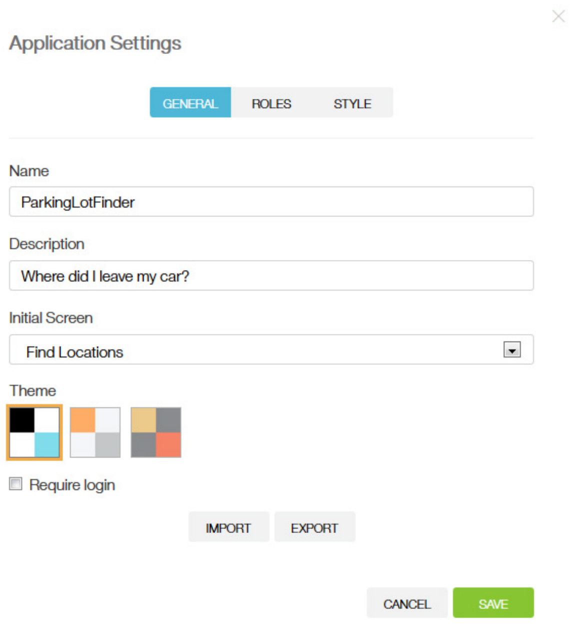
4. After a moment or two, your application should show **Running** as its status, and your application will have been deployed on Bluemix. (This is why you needed a Bluemix account to base your RapidApps account on.) Your application is now available in the cloud to users everywhere.
5. To test your application in "production," open a new browser window or tab. Enter your hostname plus the Bluemix domain `mybluemix.net`, which make up the fully qualified domain name of your application. (For example, since I entered `ParkingLotFinder` as my hostname, I would go to the URL `http://ParkingLotFinder.mybluemix.net`.)
6. Your browser should display your application. If you wish, you can test your application on the cloud. Note that there is a **Sign In** option in the right corner. This allows you to log in to your application, using either a Google, Facebook, or Bluemix ID. Your ID from the system you choose will be used in the Driver field whenever you create a record in the database.



You can now let your friends know your application is available in the cloud.

Step 9. Set up basic security

1. If multiple people will be using your cloud-based application, an easy security measure to take is to require that users log in before accessing the application. To do this, close the tab with your application, and return to your RapidApps dashboard.
2. Click the name of your application in the application's icon. This will open the application in the designer again. (This is how you would add functionality to your application, as well.)
3. In the top-right corner, click the Settings gear icon. This will display the application's basic



Name: ParkingLotFinder

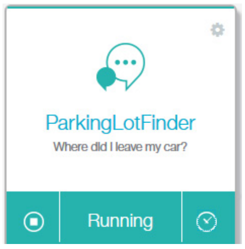
Description: Where did I leave my car?

Initial Screen: Find Locations

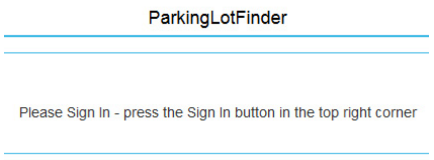
☐ Require login

settings.

4. Check the **Require login** box and press **Save**. This will require users to log in to your application before interacting with it.
5. Return to your dashboard and click the Version Control circled check mark at the bottom right side of the application's icon. This will show you the versions of your application that are stored in the RapidApps system.



6. Click the Run icon for the version marked "Currently editing." You can leave the hostname as it is. Start the application. RapidApps will stop the previous version and then deploy and start the new version.
7. Once the version is started, open a new tab or browser and enter your URL again. It did not change because you did not change the hostname.


ParkingLotFinder

Please Sign In - press the Sign In button in the top right corner

Note that users now have to log in before they can use the application. This is core RapidApps functionality; all you did was turn it on. This guarantees you will have a driver name pre-filled since users will have to login (using one of the available services) before they can save their car's location.

Conclusion

And that's it! You've built an application that allows you and other users to store a parked car's location, deployed the app to IBM Bluemix, and accessed it from the cloud.

RELATED TOPICS: RapidApps developer center RapidApps Q&A Bluemix developer center

About the author

Kevin J. Gilhooly

@xriva on Twitter
Follow me on G+
Find me on LinkedIn

© Copyright IBM Corporation 2014
(www.ibm.com/legal/copytrade.shtml)
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)