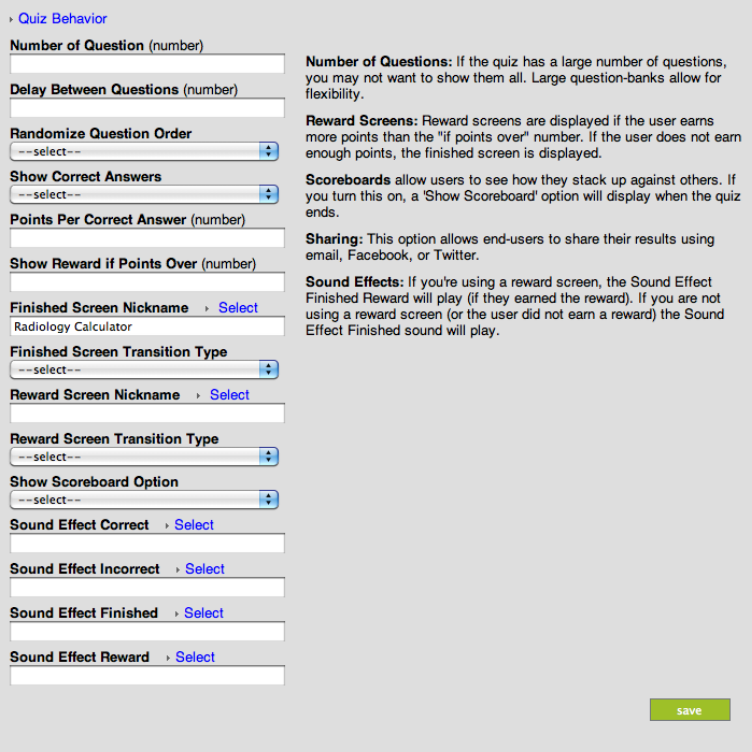
Configuring Quizzes

buzztouch v1.5

Quizzes are a fun way to communicate, gather data about app users, and spice up your application. There are lots of options to choose from when configuring quiz screens. Quiz screens share some common properties with other screen types such as background colors, navigation bar titles, etc. This document focusses on some less obvious quiz properties.

Show Correct Answers: When a user is taking a quiz, and selects and answer, you may or may not want to show the correct answer. Showing the correct answer will highlight the answer answer when they make a selection.



Quiz Behavior

Number of Question (number)

Delay Between Questions (number)

Randomize Question Order: --select--

Show Correct Answers: --select--

Points Per Correct Answer (number)

Show Reward if Points Over (number)

Finished Screen Nickname Select: Radiology Calculator

Finished Screen Transition Type: --select--

Reward Screen Nickname Select

Reward Screen Transition Type: --select--

Show Scoreboard Option: --select--

Sound Effect Correct Select

Sound Effect Incorrect Select

Sound Effect Finished Select

Sound Effect Reward Select

Number of Questions: If the quiz has a large number of questions, you may not want to show them all. Large question-banks allow for flexibility.

Reward Screens: Reward screens are displayed if the user earns more points than the "if points over" number. If the user does not earn enough points, the finished screen is displayed.

Scoreboards allow users to see how they stack up against others. If you turn this on, a 'Show Scoreboard' option will display when the quiz ends.

Sharing: This option allows end-users to share their results using email, Facebook, or Twitter.

Sound Effects: If you're using a reward screen, the Sound Effect Finished Reward will play (if they earned the reward). If you are not using a reward screen (or the user did not earn a reward) the Sound Effect Finished sound will play.

Points per Correct Answer: Use this property to set how many points a user earns when they answer a question correctly.

Show Reward if Points Over: This property is used in conjunction with a Reward Screen. The idea is that if a user answers earns enough points you may want to show them a reward screen. This means you'll need to configure a screen to show as a reward prior to setting this property. Is this feature is used, a button labeled 'Show Reward' will show when the quiz ends.

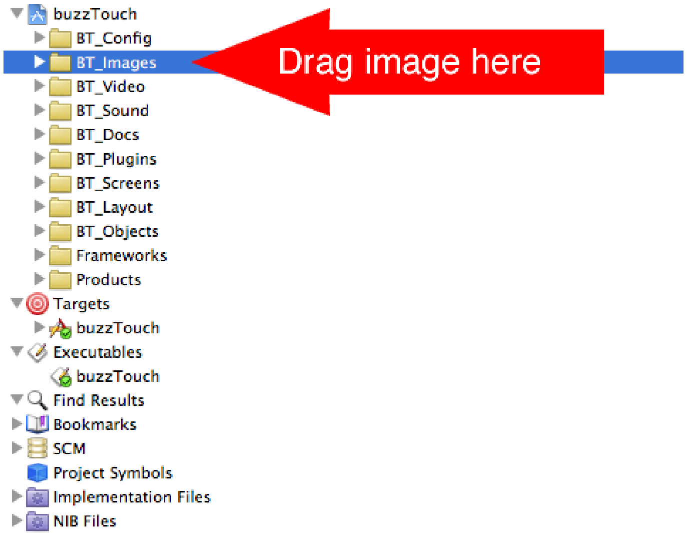
Finished Screen Nickname / Reward Screen Nickname: When a user is done taking a quiz, you may want to lead them to another screen, or, if you want, to a Reward Screen. Finish Screens and Reward Screens can be any type of screens you configure. You'll need to configure these screens prior to setting these properties. This is because you need to know the nickname of the screen to use as the finish or reward screen. If a user earns a reward, the reward screen option will show and the finish screen option will not. If the user does not earn a reward, or if no reward is configured, the finish screen option with show only if you configured one.

Question Text: Some quizzes use text-only to 'ask the question' and others use images only. And, some use both text and images. When adding or updating questions, the Question Text is required. However, you may not want to show any text. In this case, enter the word “none” (without the quotes) and the text will not display in the quiz. This is useful when you're creating quizzes that use images only.

Question Images: If you're using images for each quiz question, be sure they are the proper dimensions. Any “size” will work but they look much better if they are sized for their purpose.

Image Names: If you use image names (and not image URL's) you'll need to “add images” to your project after downloading the source-code from your buzztouch control panel. Add images to the project by dragging them into the BT_Images folder. Drag them into your project in Xcode or Eclipse (after opening your project) and not into the source-code folder you downloaded. Xcode and Eclipse need to “know about” these images and adding them to the project is how you tell Xcode or Eclipse “hey, I wanna use these images in my project” Tip: Create a folder on your computer to store all your app images. When you download the source-code for your project, open Xcode or Eclipse and drag the entire folder of images into the BT_Images folder.

Image URL's: If you use image URL's (and not image names) the images will be downloaded from the internet and saved on the device. Images are saved to improve performance and are available when the application is offline.



Drag image here

Image Sizes:
Recommended image size: 300 x 150 for small devices. 748 x 350 for large devices.

Data URL: If you want to pull questions from a remote data source, such as your web-server, enter the URL to the file. The file may be static, such as a plain text file of questions, or it may be dynamic, such as a PHP script that outputs questions stored in a database.

If you use a text file, you may enter something like this in the Data URL box:

http://www.mysite.com/quizquestions.txt

If you use a dynamic script, you may enter something like this in the Data URL box:

http://www.mysite.com/quizquestions.php

Both of these URL's need to output Quiz Questions in a valid JSON format. Each question consists of an itemId, itemType, questionText, correctAnswerText, incorrectText1, incorrectText2, and incorrectText3. Optionally, it may include an imageNameSmallDevice, imageURLSmallDevice, imageNameLargeDevice and imageURLLargeDevice.

A series of question items is wrapped in a “childItems” object and output like this:

```
{"childItems":[

        {"itemId":"question1", "itemType":"BT_questionItem", "questionText":"What is the
        capital city of Nevada?", "correctAnswerText":"Carson City", "incorrectText1":"Las Vegas",
        "incorrectText2":"Reno", "incorrectText3":"Tahoe",
        “imageURLSmallDevice”:”http://www.mywebsite.com/quizImage1.png”},

        {"itemId":"question2", "itemType":"BT_questionItem", "questionText":"What is the
        capital city of California?", "correctAnswerText":"Sacramento", "incorrectText1":"Modesto",
        "incorrectText2":"Monterey", "incorrectText3":"Los Angeles",
        “imageURLSmallDevice”:”http://www.mywebsite.com/quizImage2.png”}

]
}
```

Pay special attention to the square brackets. All the questions items are inside the square brackets. Also, pay special attention to the commas separating each question. The last question in the list is not followed by a comma. The white-space you see between questions should not be included in the output, it is here for readability only.

Scoreboard URL: Scoreboard URL's serve two purposes. First, they “accept” quiz results so they can be saved to a backend database. Next, they output JSON data that is consumed by the mobile application to produce the scoreboard screen. This means the script at this URL has two jobs. To save quiz results, and to output quiz results. Quiz Results are appended to the end of the URL. Example, if you setup a custom script at:

http://www.mywebsite.com/quizresults.php?command=getQuizResults

The mobile application would append variables to the end of the URL and request this URL:

http://www.mywebsite.com/quizresults.php?command=getQuizResults &VARIABLES HERE

These variables will be appended to the URL:

NOTE: Your URL must include the starting question mark (like the example above). If you don't have an important variable to append, use quizresults.php?1=1 or anything else. The mobile app will append ampersands and values so you need to append the first question mark.

userDisplayName=the name entered in the score-board screen name box (string)
totalPoints=quiz points earned (integer)
totalSeconds=quiz seconds to complete (integer)
numberOfQuestions=quiz number of questions (integer)
numberCorrect=quiz number of questions answered correctly (integer)
numberIncorrect=quiz number of questions answered incorrectly (integer)
deviceModel=the device model (String, ex: iPhone, iPod Touch, Motorola Droid)
deviceId=the unique device id (String, usually around 20 characters but could be up to 50)
deviceLatitude=the latitude of the device (float, may be negative, may be blank)
deviceLongitude=the longitude of the device (float, may be negative, may be blank)

Scoreboard Output: After your URL accepts the data, and saves it, it will need to output JSON data to return to the mobile device. This data is used to display the scoreboard list. A series of scoreboard items is wrapped in a “childItems” object and output like this:

```
{"childItems":[


        {"itemId":"1", "titleText":"Joe", "descriptionText":"03-20-2011 13:25:23 (gmt) 750 points in 60 sec. Ipad-
        Simulator", "rowAccessoryType":"none", "loadScreenWithItemId":"none"},


        {"itemId":"2", "titleText":"Bill", "descriptionText":"03-20-2011 13:26:35 (gmt) 750 points in 14 sec. Ipad-
        Simulator", "rowAccessoryType":"none", "loadScreenWithItemId":"none"},


        {"itemId":"3", "titleText":"Sally", "descriptionText":"03-20-2011 13:32:38 (gmt) 750 points in 323 sec.
        iPhone-Simulator", "rowAccessoryType":"none", "loadScreenWithItemId":"none"},


        {"itemId":"4", "titleText":"Tina", "descriptionText":"03-20-2011 13:08:48 (gmt) 150 points in 5 sec. Iphone-
        Simulator", "rowAccessoryType":"none", "loadScreenWithItemId":"none"}


]
}
```

Like before, the white-space you see between items should not be included in the output, it is here for readability only.

NOTE: It's important to include “none” for the rowAccessoryType and “none” for the loadScreenWithItemId values so nothing happens when the item is selected in the list. Yes, you could customize this behavior. The scoreboard is a BT_screen_menuList so you can include any properties any other menu-list may use. That discussion is beyond the scope of this document.