Titanium Certified Expert (TCE)
Certification Objectives

Revision: September 2013

TCE Overview

Titanium Certified Expert (TCE) is Appcelerator's second-level certification. Prospective candidates must have earned their TCD certification before they will be eligible to earn this certification. Successful candidates will have the skills and knowledge to create "best of breed" mobile applications that implement best practices and take advantage of a wide variety of cross-platform and platform-specific APIs.

The knowledge/skill domain section of this document describes the concepts and skills that a candidate should possess prior to attempting the TCE exam. This document is not a comprehensive listing of all content of the exam. Candidates should use this document as a guide to their learning.

The TCE exam includes 66 randomly selected questions. Candidates must earn a grade of 75% or higher to pass. Question types include: true/false, multiple-choice, multiple-select, and fill-in-the-blank. Candidates might be asked to review code for purpose or correctness, but will not be asked to write code as part of the exam.

Appcelerator regularly reviews the content of our exams and updates questions or exams as Titanium software or training course changes warrant. When necessary, we will publish updated certification objectives and/or exams.

Knowledge/Skill Domains

A successful candidate will be prepared to answer questions that demonstrate skill and understanding of each of the following knowledge domains:

1. UI/UX design principles
    a. Identify appropriate user experience (UX) and UI design criteria for mobile apps
    b. Identify the key components of a compelling UI design
    c. Identify common tools and techniques used to create UX and UI designs
    d. Identify platform-specific app design considerations, such as vendor-specific interface guidelines, UI/UX conventions, and user expectations that you must follow when creating apps for that platform
    e. Identify strategies for addressing platform differences in the UX/UI of an app

2. Orientation and gestures
    a. Lock orientation so that an app's screens do not rotate with the device
    b. Handle rotation so that an app's screens rotate and the UI redraws appropriately with the device's orientation
    c. Handle non-tap events, such as swipe, longpress, and shake
    d. Use command-keys to rotate the iOS simulator and Android emulator

3. Animation
    a. Animate the basic properties of a UI element
    b. Create and apply 2D and 3D transformations of UI elements
    c. Implement iOS-specific transitions on windows and views

   d. Implement Alloy animation built-ins

4. TableViews and ListViews
   a. Implement TableViews and TableViewRows to display list-like data in your app.
   b. Implement ListViews, ListTemplates, ListSections, ListItems, and ListDataItems to display list-like data in your app.
   c. Handle user-generated events with both TableViews and ListViews.
   d. Identify the similarities and differences between TableViews and ListViews.
   e. Compare and contrast the benefits of using TableViews, ListViews, or ScrollViews to display list-like content in your app.
5. User input and navigation
   a. Implement cross-platform and platform-specific features of common user input components, such as buttons, text fields, and textareas
   b. Implement UI controls, such as pickers, options dialogs, and tabbed bars, that are platform-specific or have platform-specific options
   c. Handle the soft keyboard using cross-platform and platform-specific techniques
   d. Set the keyboard type and return button type
   e. Handle events associated with user input components

6. Custom UI components
   a. Describe the building blocks and techniques for creating custom UI components
   b. Implement an Alloy widget in a Titanium Mobile app
   c. Create an Alloy widget and use it in a project
   d. Identify sources from which you can download Widgets

7. iOS platform specifics
   a. Identify and implement iOS-only, iPhone-only, and iPad-only user interface components
   b. Identify and implement iOS-only functional (non-UI) Titanium APIs
   c. Configure platform-specific app options via the tiapp.xml and Info.plist files
   d. Enable user-configurable application settings using a Settings bundle
   e. Implement navigation controls and app architecture according to platform best-practices

8. Android platform specifics
   a. Identify and implement Android-only user interface components
   b. Identify and implement Android-only functional (non-UI) Titanium APIs
   c. Describe and implement Android components and terminology, such as Activities and Intents
   d. Configure platform-specific app options via the tiapp.xml and AndroidManifest.xml files
   e. Configure platform-specific app options in that platform's native configuration file
   f. Implement navigation controls and app architecture according to platform best-practices

9. Application interconnections
   a. Enable your apps to launch other apps on the user's device or connect the user to the appropriate app store to download the target app
   b. Configure app startup parameters and implement custom URL schemes so that other apps can launch yours
   c. Pass data to apps you launch via arguments

10. Performance, optimization, and best practices
    a. Identify the primary bottlenecks in a mobile app and describe the techniques used to alleviate problems
    b. Implement memory management techniques
    c. Prevent memory leaks when implementing JavaScript coding techniques, Titanium APIs, and event listeners
    d. Identify database, networking, and device hardware optimization techniques
    e. Identify JavaScript best practices for Titanium apps
    f. Identify Titanium API best practices (e.g. how to optimize TableView performance)

11. Titanium extension modules
    a. Install and use a module within a Titanium app
    b. Identify sources for free and commercial Titanium modules
    c. Describe the distribution and support models for Titanium extension modules
    d. Identify module development documentation and reference sources