Titanium Certified Developer (TCD)
Certification Objectives

Revision: February 2013

TCD Overview

Titanium Certified Developer (TCD) is Appcelerator's first-level application developer certification. Successful candidates will have the skills and knowledge to create native mobile applications using the Titanium framework. Candidates should be familiar with JavaScript and comfortable writing intermediate level JavaScript-based scripts for the web or other environments.

The knowledge/skill domain section of this document describes the concepts and skills that a candidate should possess prior to attempting the TCD 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 TCD exam includes 65 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. JavaScript fundamentals
    a. Understand JavaScript syntax and fundamentals
    b. Identify CommonJS coding patterns
    c. Describe JavaScript variable handling topics, such as scope, hoisting, data type conversions,
    d. Describe JavaScript function topics, such as closures, arguments handling, the keyword 'this', and pass by value.
    e. Describe JavaScript object topics, such as prototypes, prototypal inheritance, and parasitic inheritance.

2. Titanium basics
    a. Create a Titanium project using both Studio and the Command Line Interface
    b. Run a Titanium project in the simulator/emulator
    c. Configure app properties such as the SDK version, target platforms, etc.
    d. Describe the architecture of Titanium
    e. Describe execution context and the means by which you create one or more

3. User interface
    a. Select the appropriate UI measurement units
    b. Position elements on screen accounting for the UI coordinates system

c. Select and implement layout modes
d. Fire and react to user and non-user events
e. Specify the app icon and splash screens used by your app.
f. Include platform-specific resources at build time
g. Include density and aspect-ratio specific images at build time
h. Internationalize your app using replaceable strings.

4. Alloy
    a. Describe the role and proper syntax of Alloy Views, Styles, Controllers, Model, and Collections
    b. Implement the Titanium components that comprise an app's UI by using Alloy
    c. Handle user interaction and app-level events in your controllers
    d. Implement media/device queries to target Alloy styles to a specific platform or form factor
    e. Represent your app's data in Models and Collections
    f. Bind Models and Collections to Views
    g. Save data locally via sync adapters
    h. Create an Alloy Migration and manage database versioning

5. Networking
    a. Implement the HTTPClient object
    b. Retrieve data in various formats from network services
    c. Upload and download files across the network
    d. Post JSON-formatted data to a web service; receive and process JSON data from a network endpoint
    e. Receive and process XML data from a network endpoint
    f. Handle network and data errors

6. Multimedia
    a. Display still images as foreground and background graphics
    b. Implement local and streaming audio playback
    c. Implement local and streaming video playback
    d. Capture still images and video from the camera
    e. Retrieve images from the device’s photo gallery app

7. Local data storage
    a. Identify the database capabilities of the mobile operating environments
    b. Store data in an application property
    c. Retrieve data from an application property
    d. Determine when app properties are the most suitable storage location for your app’s data

8. Filesystem
    a. Store text and binary data in files
    b. Read text and binary data from files
    c. Identify the accessible storage locations on the filesystem
    d. Identify the appropriate locations to store data on the filesystem
    e. Determine when the filesystem is the most suitable storage location for your app’s data

9. Geolocation and mapping
    a. Configure geolocation on Android, iOS, and Mobile Web
    b. Request geolocation permissions, accounting for platform-specific requirements
    c. Obtain the user's current location and/or continually monitor the user's location
    d. Perform forward and reverse geocoding
    e. Add a map to your app
    f. Set map options and properties
    g. Add annotations to your map, and set annotation options and properties
    h. Enable event handling for maps and annotations

10. Appcelerator Cloud Services
    a. Identify ACS APIs and features
    b. Describe ACS security features
    c. Cloud-enable an app and manage app keys
    d. Manage user security within your ACS app
    e. Store data in and retrieve data from the ACS cloud
    f. Identify the features of and use cases for NodeACS

11. Web content
    a. Include HTML/CSS content in your Titanium app
    b. Use meta tags to control the canvas size and zoom scale of a WebView
    c. Identify the ramifications and pitfalls of the WebView
    d. Communicate between the WebView and native Titanium environments

12. Debugging and tools
    a. Identify the Titanium Studio features you can use to debug an application
    b. Use breakpoint debugging with Studio
    c. Identify the troubleshooting and development tools provided by the Android and iOS SDKs that you will use as a Titanium developer
    d. Identify the troubleshooting tools and techniques that you will use when debugging Mobile Web apps

13. Deployment and publishing
    a. Identify the certificate types and keys you will need to publish an app for testing or final distribution
    b. Code-sign your app
    c. Install an iOS app for testing
    d. Install an Android app for testing
    e. Launch a Mobile Web app for testing
    f. Identify the requirements and procedure for publishing an app to the Android market (Google Play), iTunes app store, and a Mobile Web app to the Internet
    g. Identify the requirements and procedure for publishing an iOS app to non-market locations (Enterprise, ad-hoc, etc.)