# Chapter -1 Introduction to Hybrid application, development platforms

## 1. What is hybrid application? Need of hybrid application development

A hybrid application (hybrid app) is one that combines elements of both native and Web applications. Native applications are developed for a specific platform and installed on a computing device. Web applications are generalized for multiple platforms and not installed locally but made available over the Internet through a browser. Hybrid apps are often mentioned in the context of mobile computing.

Hybrid application features:

- Can function whether or not the device is connected.
- Integration with a device's file system.
- Integration with Web-based services.
- An embedded browser to improve access to dynamic online content.

Most applications could be considered hybrid apps. Web apps, such as online banking services, typically store some content locally; locally stored native apps, such as Microsoft Word, also interface to the Internet.

## 2. Tool and platforms in used for development of hybrid mobile application development

### *(1) Phonegap-cordova*

1. Apache Cordova is a mobile application development framework originally created by Nitobi. Adobe Systems purchased Nitobi in 2011, rebranded it as PhoneGap, and later released an open source version of the software called Apache Cordova.

2. Apache Cordova enables software programmers to build applications for mobile devices using CSS3, HTML5, and JavaScript instead of relying on platform-specific APIs like those in Android, iOS, or Windows Phone. It enables wrapping up of CSS, HTML, and JavaScript code depending upon the platform of the device. It extends the features of HTML and JavaScript to work with the device.

3. The resulting applications are hybrid, meaning that they are neither truly native mobile application (because all layout rendering is done via Web views instead of the platform's native UI framework) nor purely Web-based (because they are not just Web apps, but are packaged as apps for distribution and have access to native device

APIs). Mixing native and hybrid code snippets has been possible since version 1.9.

***(2) Ionic***

1. Ionic Framework is an open source UI toolkit for building preferment, high-quality mobile and desktop apps using web technologies (HTML, CSS, and JavaScript).
2. Ionic Framework is focused on the frontend user experience, or UI interaction of an app (controls, interactions, gestures, animations). It's easy to learn, and integrates nicely with other libraries or frameworks, such as Angular, or can be used standalone without a frontend framework using a simple script include.

***(3) Mobile angular UI***

1. Mobile Angular UI provides essential mobile components that are missing in Bootstrap 3: switches, overlays, sidebars, scrollable areas, absolute positioned top and bottom navbars that don't bounce on scroll.
2. It relies on robust libraries like fastclick.js and overthrow.js to achieve the better mobile experience.

## 3. Step by step installation of coredova using git and npm

**Steps:**

1. Make sure you have an up-to-date version of Node.js installed on your system. If you don't have Node.js installed, you can install it from here.
2. Open a terminal window (Mac) or a command window (Windows), and install Cordova and Ionic:

```
npm install -g cordova ionic
```

3. On a Mac, you may have to use sudo depending on your system configuration:

```
sudo npm install -g cordova ionic
```

4. If you already have Cordova and Ionic installed on your computer, make sure you update to the latest version:

```
npm update -g cordova ionic
```

**or**

```
sudo npm update -g cordova ionic
```

## 4. Introduction to HTML 5 and HTML 5 APIs

***(1) <u>Forms validation</u>***

- ✓ Form validation helps us to ensure that users fill out forms in the correct format, making sure that submitted data will work successfully with our applications.

**<u>There are two different types of form validation which you'll encounter on the web:</u>**

- **<u>Client-side</u>** <u>validation</u> is validation that occurs in the browser before the data has been submitted to the server. This is more user-friendly than server-side validation as it gives an instant response.

  ***<u>This can be further subdivided:</u>***

  - ➢ **<u>JavaScript validation</u>** is coded using JavaScript. It is completely customizable.
  - ➢ **<u>Built-in form validation</u>** using HTML5 form validation features. This

- ➢ generally does not require JavaScript. Built-in form validation has better performance, but it is not as customizable as JavaScript.

- **Server-side validation** is validation which occurs on the server after the data has been submitted. Server-side code is used to validate the data before it is saved into the database. If the data fails authentication, a response is sent back to the client to tell the user what corrections to make.
  Server-side validation is not as user-friendly as client-side validation

## *(2) Audio video tags*

- HTML5 features include native audio and video support without the need for Flash.
- The HTML5 <audio> and <video> tags make it simple to add media to a website. You need to set src attribute to identify the media source and include a controls attribute so the user can play and pause the media.

Here is the simplest form of embedding a video file in your webpage –

```
<video src = "foo.mp4” width = "300" height = "200" controls>
  Your browser does not support the <video> element.
</video>
```

## Video Attribute Specification

**The HTML5 video tag can have a number of attributes to control the look and feel and various functionalities of the control –**

| Sr.No. | Attribute & Description |
|---|---|
| 1 | **autoplay**<br>This Boolean attribute if specified, the video will automatically begin to play back as soon as it can do so without stopping to finish loading the data. |
| 2 | **autobuffer**<br>This Boolean attribute if specified, the video will automatically begin buffering even if it's not set to automatically play. |

|  |  |
|---|---|
| 3 | **controls**<br><br>If this attribute is present, it will allow the user to control video playback, including volume, seeking, and pause/resume playback. |
| 4 | **height**<br><br>This attribute specifies the height of the video's display area, in CSS pixels. |
| 5 | **loop**<br><br>This Boolean attribute if specified, will allow video automatically seek back to the start after reaching at the end. |
| 6 | **preload**<br><br>This attribute specifies that the video will be loaded at page load, and ready to run. Ignored if autoplay is present. |
| 7 | **poster**<br><br>This is a URL of an image to show until the user plays or seeks. |
| 8 | **src**<br><br>The URL of the video to embed. This is optional; you may instead use the <source> element within the video block to specify the video to embed. |
| 9 | **width**<br><br>This attribute specifies the width of the video's display area, in CSS pixels. |

## Embedding Audio

**HTML5 supports <audio> tag which is used to embed sound content in an HTML or XHTML document as follows.**

```
<audio src = "foo.wav" controls autoplay>
   Your browser does not support the <audio> element.
</audio>
```

**Audio Attribute Specification**

**The HTML5 audio tag can have a number of attributes to control the look and feel and various functionalities of the control −**

| Sr.No. | Attribute & Description |
|---|---|
| 1 | **autoplay**<br><br>This Boolean attribute if specified, the audio will automatically begin to play back as soon as it can do so without stopping to finish loading the data. |
| 2 | **autobuffer**<br><br>This Boolean attribute if specified, the audio will automatically begin buffering even if it's not set to automatically play. |
| 3 | **controls**<br><br>If this attribute is present, it will allow the user to control audio playback, including volume, seeking, and pause/resume playback. |
| 4 | **loop**<br><br>This Boolean attribute if specified, will allow audio automatically seek back to the start after reaching at the end. |
| 5 | **preload**<br><br>This attribute specifies that the audio will be loaded at page load, and ready to run. Ignored if autoplay is present. |
| 6 | **src**<br><br>The URL of the audio to embed. This is optional; you may instead use the <source> element within the video block to specify the video to embed. |

## *(3) <u>Data storage APIs</u>*

### Local storage

- With web storage, web applications can store data locally within the user's browser.
- Before HTML5, application data had to be stored in cookies, included in every server request. Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

- Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.
- Web storage is per origin (per domain and protocol). All pages, from one origin, can store and access the same data.

**Web sql**

Web SQL Database is a web page API for storing data in databases that can be queried using a variant of SQL. The API is supported by Google Chrome, Opera, Safari and the Android Browser.

**IndexedDB**

IndexedDB is a low-level API for client-side storage of significant amounts of structured data, including files/blobs. This API uses indexes to enable high-performance searches of this data.

## 5. Introduction to CSS, Sscss,

## less CSS

- **CSS - CSS** stands for **C**ascading **S**tyle **S**heets
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External style sheets are stored in **CSS**

**files Sass**

Sass Variables allow you to define a value once and use it in multiple places. Variables begin with dollar
signs and are set like CSS properties. You can change the value of the variable in one place, and all instances where it is used will be changed, too. For example, if you wanted to set the same height for two different selectors, you could create a variable called $control-height:

```
$control-height: 40px;
```

**Less**

Less (which stands for Leaner Style Sheets) is a backwards-compatible language extension for CSS. This is the official documentation for less, the language and Less.js, the JavaScript tool that converts your Less styles to CSS styles. Because less looks just like CSS, learning it is a breeze.

# CHAPTER -2 Java Script for Mobile Application Development

## 1. Introduction to JavaScript

- JavaScript is the Programming Language for the Web
- JavaScript can update and change both HTML and CSS
- JavaScript can calculate, manipulate and validate data

## JavaScript Variables

**JavaScript variables are containers for storing data**

**values. var x = 5;**
**var y = 6;**
**var z = x + y;**

**From the example above, you can expect:**

**x stores the value**
**5 y stores the**
**value 6 z stores**
**the value 11**

## 2. What is jQuery?

- JQuery is a lightweight, "write less, do more", and JavaScript library.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- JQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- JQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

**The jQuery library contains the following features:**

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

## 3. What is AngularJS?

AngularJS is an open source Model-View-Controller framework which is similar
to the JavaScript framework.

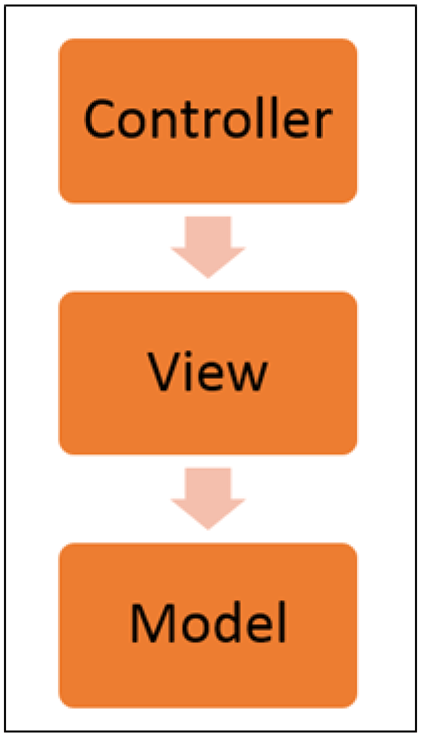
### AngularJS Features

Angular has the following key features which makes it one of the powerful frameworks in the market.

1. MVC **– The framework is built on the famous concept of MVC (Model-View-Controller). This is a design pattern used in all modern day web applications. This pattern is based on splitting the business logic layer, the data layer, and presentation layer into separate sections. The division into different sections is done so that each one could be managed more easily.**
2. Data Model Binding **– You don't need to write special code to bind data to the HTML controls. This can be done by Angular by just adding a few snippets of code.**
3. Writing less code **– When carrying out DOM manipulation a lot of JavaScript was required to be written to design any application. But with Angular, you will be amazed with the lesser amount of code you need to write for DOM manipulation.**
4. Unit **Testing ready – The designers at Google not only developed Angular but also developed a testing framework called "Karma" which helps in designing unit tests for AngularJS applications.**

## 4. AngularJS Architecture

**Angular.js follows the MVC architecture, the diagram of the MVC framework as shown below.**



**AngularJS Architecture Diagram**

## 5. Iconic

Ionic Framework is an open source UI toolkit for building performant, high-quality mobile and desktop apps using web technologies (HTML, CSS, and JavaScript).

Ionic Framework is focused on the frontend user experience, or UI interaction of an app (controls, interactions, gestures, animations). It's easy to learn, and integrates nicely with other libraries or
frameworks, such as Angular, or can be used standalone without a frontend framework using a simple script include.

### Goals

1. Cross-platform
    a. Build and deploy apps that work across multiple platforms, such as native iOS, Android, desktop, and the web as a Progressive Web App - all with one code base. Write once, run anywhere.
2. Web Standards-based
    a. Ionic Framework is built on top of reliable, standardized web technologies: HTML, CSS, and JavaScript, using modern Web APIs such as Custom Elements and Shadow DOM. Because of this, Ionic components have a stable API, and aren't at the whim of a single platform vendor.
3. Beautiful Design
    a. Clean, simple, and functional. Ionic Framework is designed to work and display beautifully out-of-the-box across all platforms. Start with pre-designed components, typography, interactive paradigms, and a gorgeous (yet extensible) base theme.
4. Simplicity
    a. Ionic Framework is built with simplicity in mind, so that creating Ionic apps is enjoyable, easy to learn, and accessible to just about anyone with web development skills.

## 6. $scope and $rootScope

- $scope is an object that is accessible from current component e.g. Controller, Service only.

- $rootScope refers to an object which is accessible from everywhere of the application. You can think $rootScope as global variable and $scope as local variables.

## 7. Config () and Run ()

- Configuration blocks (registered with module.config ()) get executed during provider registration, and can only be injected providers and constants (see module.provider () and module.constant ()). This is typically where you would configure application- wide stuff, such as the $routeProvider. Stuff that needs to be configured before the services is created.

- Run blocks (registered with module.run ()) get executed after the injector has all the providers. Now, all instances and constants can be injected. This is typically where you would configure services, $rootScope, events and so on.

## 8. Directives in AngularJS

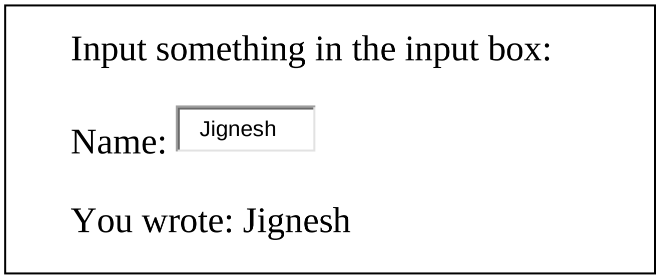
- AngularJS lets you extend HTML with new attributes called Directives.
- AngularJS has a set of built-in directives which offers functionality to your applications.
- AngularJS also lets you define your own directives.

**AngularJS directives are extended HTML attributes with the prefix ng-.**

- The ng-app directive initializes an AngularJS application.
- The ng-init directive initializes application data.
- The ng-model directive binds the value of HTML controls (input, select, text area) to application data.

```
<! DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>
<div ng-app="" ng-init="firstName='Jignesh'">
<p>Input something in the input box :</p>
<p>Name: <input type="text" ng-model="firstName"></p>
<p>You wrote: {{ firstName }}</p>
</div>
</body>
</html>
```

**OUTPUT:**



Input something in the input box:

Name: Jignesh

You wrote: Jignesh

## 9. Filters in AngularJS

**AngularJS provides filters to transform data:**

- ✓ Currency Format a number to a currency format.
- ✓ Date Format a date to a specified format.
- ✓ Filter Select a subset of items from an array.
- ✓ Json Format an object to a JSON string.
- ✓ Limit To Limits an array/string, into a specified number of elements/characters.
- ✓ Lowercase Format a string to lower case.
- ✓ Number Format a number to a string.
- ✓ OrderBy Orders an array by an expression.
- ✓ Uppercase Format a string to upper case.

## 10. Angular forms

Angular provides two different approaches to handling user input through forms: reactive and template-driven. Both capture user input events from the view, validate the user input, create a form model and data model to update, and provide a way to track changes.

- Reactive forms are more robust: they're more scalable, reusable, and testable. If forms are a key part of your application, or you're already using reactive patterns for building your application, use reactive forms.
- Template-driven forms are useful for adding a simple form to an app, such as an email list signup form. They're easy to add to an app, but they don't scale as well as reactive forms. If you have very basic form requirements and logic that can be managed solely in the template, use template-driven forms.

## 11. Angular validation

To add validation to a template-driven form, you add the same validation attributes as you would with native HTML form validation. Angular uses directives to match these attributes with Validator functions in the framework.

Every time the value of a form control changes, Angular runs validation and generates either a list of validation errors, which results in an INVALID status, or null, which results in a VALID status.

## 12. Angular module

The angular.module is a global place for creating, registering and retrieving AngularJS modules. All modules (AngularJS core or 3rd party) that should be available to an application must be registered using this mechanism.

## 13. Angular controller

- AngularJS controllers **control the data** of AngularJS applications.
- AngularJS controllers are regular **JavaScript Objects**.
- The **ng-controller** directive defines the application controller.

```
<! DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>
<body>

<div ng-app="myApp" ng-controller="myCtrl">

First Name: <input type="text" ng-
model="firstName"><br> Last Name: <input
type="text" ng-model="lastName"><br>
<br>
Full Name: {{firstName + " " + lastName}}
</div>
<script>
var app = angular.module('myApp', []);
app.controller('myCtrl',
function($scope) {
  $scope.firstName = "jignesh";
  $scope.lastName = "thanki";
});
</script>

</body>
</html>
```

**OUTPUT:-**

```
First    [jignesh]
Name:    [thanki ]
Last
Name:
```

**Full Name: jignesh thanki**

## 14. Angular factory

Angular provides us with three ways to create and register our own service.

1. Factory
2. Service
3. Provider

1. When you're using a **Factory** you create an object, add properties to it, then return that same object. When you pass this service into your controller, those properties on the object will now be available in that controller through your factory.
2. When you're using **Service**, it's instantiated with the 'new' keyword. Because of that, you'll add properties to 'this' and the service will return 'this'. When you pass the service into your controller, those properties on 'this' will now be available on that controller through your service.
3. **Providers** are the only service you can pass into your .config () function. Use a provider when you want to provide module-wide configuration for your service object before making it available.

## 15. Ui-router

What is AngularUI **Router**? The **UI**-**Router** is a **routing** framework for **AngularJS** built by the AngularUI team. It provides a different approach than ngRoute in that it changes your application views based on state of the application and not just the route URL.

## 16. $state

$state service is responsible for representing states as well as transitioning between them. It also provides interfaces to ask for current state or even states you're coming from.

## 17. $statParams and $state.params

With ui-router, it's possible to inject either $state or $stateParams into a controller to get access to parameters in the URL. However, accessing parameters through $stateParams only exposes parameters belonging to the state managed by the controller that accesses it, and its parent states,
while $state.params has all parameters, including those in any child states.

**Example and output is given in comment**

```
$stateProvider.state('a', {
   url:
   'path/:id/:anotherParam/',
   controller: 'ACtrl',
 });

$stateProvider.state('a.b',
   {                          url:
   '/:yetAnotherParam',
   controller: 'ABCtrl',
 });
module.controller('ACtrl', function($stateParams, $state) {
  $state.params; // has id, anotherParam, and yetAnotherParam
  $stateParams; // has id and anotherParam
}

module.controller('ABCtrl', function($stateParams, $state) {
  $state.params; // has id, anotherParam, and yetAnotherParam
  $stateParams;  // has id, anotherParam, and yetAnotherParam
}
```

## 18. $statProvider.stat ()

$stateProvider allows us to give names for routes. Having a name we can duplicate the route with another name assign different controller, view, well.. we can do whatever we want!

It means… with this approach we have different states of the route, that's why it's called **$stateProvider**.

```
$stateProvider
        .state("contact",
          { url:
          "/contact/",
          templateUrl:
```

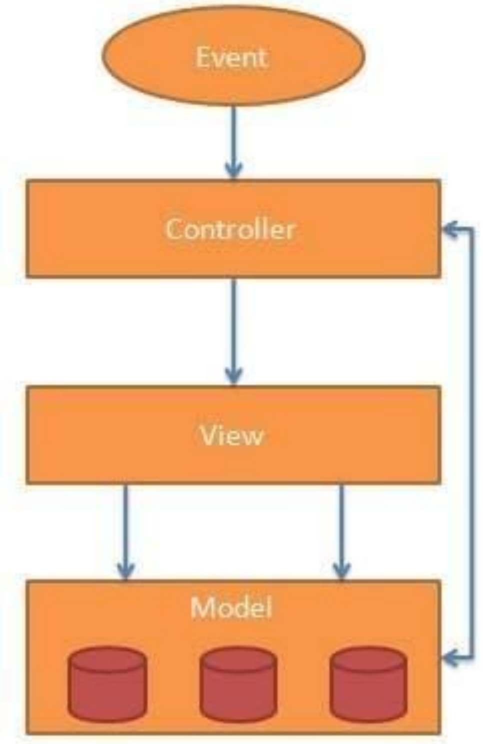
```
    '/app/Aisel/Contact/views/contact.html',
    controller: 'ContactCtrl'
});
```

## 19. MVC architecture of AngularJS

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts −

- **Model –** It is the lowest level of the pattern responsible for maintaining data.
- **View –** It is responsible for displaying all or a portion of the data to the user.
- **Controller** − It is a software Code that controls the interactions between the Model and View.

MVC is popular because it isolates the application logic from the user interface layer and supports separation of concerns. The controller receives all requests for the application and then works with the model to prepare any data needed by the view. The view then uses the data prepared by the controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows**.**



## 20. Controller and view for data handling

**The View**

A presentation of data in a particular format, triggered by the controller's decision to present the data. They are script-based template systems such as JSP, ASP, PHP and very easy to integrate with AJAX technology.

**The Controller**

The controller responds to user input and performs interactions on the data model objects. The controller receives input, validates it, and then performs business operations that modify the state of the data model.

# CHAPTER -3 Ionic - 1

## 1. Component of iconic-1

### 1) Colors

Ionic framework gives us a set of **nine predefined color classes**. You can use these colors or you can override it with your own styling.

The following table shows the default set of nine colors provided by Ionic. We will use these colors for styling different Ionic elements in this tutorial. For now, you can check all the colors as shown below –

| Class | Description | Result |
|---|---|---|
| light | To be used for white color | |
| stable | To be used for light grey color | greay |
| positive | To be used for blue color | 🟦 |
| calm | To be used for light blue color | 🟦 |
| balanced | To be used for green color | 🟩 |
| energized | To be used for yellow color | 🟨 |
| assertive | To be used for red color | 🟥 |
| royal | To be used for violet color | 🟪 |
| dark | To be used for black color | ⬛ |

```
<div class = "bar bar-header bar-calm">
  ...
</div>
```

## 2) header

Headers are fixed regions at the top of a screen that can contain a title label, and left/right buttons for navigation or to carry out various actions.
Headers come in a variety of default color options:

**Bar – light**

```
<div class="bar bar-header bar-light">
  <h1 class="title">bar-light</h1>
</div>
```

**Bar – stable**

```
<div class="bar bar-header bar-stable">
  <h1 class="title">bar-stable</h1>
</div>
```

**Ref:-** https://ionicframework.com/docs/v1/components/#header

## 3) button

The Button, an essential part of any mobile experience. Like the Header, they come in the full spectrum of Ionic's default colors.

By default a button has display: inline-block applied. Other options include block buttons for a full width.

```
<button class="button">
  Default
</button>

<button class="button button-light">
  button-light
</button>

<button class="button button-stable">
  button-stable
</button>

<button class="button button-positive">
  button-positive
</button>

<button class="button button-calm">
```

```
  button-calm
</button>

<button class="button button-balanced">
  button-balanced
</button>

<button class="button button-energized">
  button-energized
</button>

<button class="button button-assertive">
  button-assertive
</button>

<button class="button button-royal">
  button-royal
</button>

<button class="button button-dark">
  button-dark
</button>
```

**Ref:-** https://ionicframework.com/docs/v1/components/#buttons

## 4) list

The List is a common and simple way of displaying... that's right, a list. This is a widely used interface across most current mobile OS's, and can include content ranging from basic text all the way to buttons, toggles, icons, and thumbnails.

The list view is a very versatile and powerful component. List views support various interaction modes such as editing, swipe to edit, drag to reorder, and pull to refresh.

```
<ul class="list">
    <li class="item">
      ...
    </li>
</ul>
```

## 5) card

Cards have become widely used in recent years. They are a great way to contain and organize information, while also setting up predictable expectations for the user. With so much content to display at once, and often so little screen realestate, cards have fast become the design pattern of choice for many companies, including the likes
of Google, Twitter, and Spotify..

For mobile experiences, Cards make it easy to display the same information visually across many different screen sizes. They allow for more control, are flexible, and can even be animated. Cards are usually placed on top of one another, but they can also be used like a "page" and swiped between, left and right.

```
<div class="card">
  <div class="item item-text-wrap">
    This is a basic Card which contains an item that has wrapping text.
  </div>
</div>
```

**Ref-** https://ionicframework.com/docs/v1/components/#card-headers-footers

## 6) form

A form contains Placeholder Labels, Inline Label, Stacked Labels, Floating Labels, Inset forms, Inset Inputs, Input Icons, Header Inputs, Toggle, Checkbox, Radio buttons, Range, Select, Tabs ,Grid, Utility.

## 7) Checkbox

Checkboxes allow the user to select a number of items in a set of choices. A good use for a checkbox list would be a filter list to apply multiple filters to a search.

Checkboxes can also have colors assigned to them, such as checkbox-assertive to assign the assertive color.

```
<ion-list>

  <ion-checkbox ng-model="filter.red">Red</ion-checkbox>
  <ion-checkbox ng-model="filter.yellow">Yellow</ion-checkbox>
  <ion-checkbox ng-model="filter.pink">Pink</ion-checkbox>

</ion-list>
```

## 8) radio button

Radio buttons let the user select one option in a set of options, unlike a checkbox which allows for multiple selections.

```
<ion-list>
  <ion-radio ng-model="choice" ng-value="'A'">Choose A</ion-radio>
  <ion-radio ng-model="choice" ng-value="'B'">Choose B</ion-radio>
</ion-list>
```

## 9) range

This is a Range. Ranges can be themed to any default Ionic color, and used in various other elements such as a list item or card.

```
<div class="item range">
  <i class="icon ion-volume-low"></i>
  <input type="range" name="volume">
  <i class="icon ion-volume-high"></i>
</div>

<div class="list">
  <div class="item range range-positive">
    <i class="icon ion-ios-sunny-outline"></i>
    <input type="range" name="volume" min="0" max="100" value="33">
    <i class="icon ion-ios-sunny"></i>
  </div>
</div>
```

## 10) select

Ionic's select is styled so its appearance is prettied up relative to the browser's default style. However, when the select elements are opened, the default behavior on how to select one of the options is still managed by the browser.

Each platform's user-interface will be different as the user is selecting an option. For example, on a desktop browser you'll see the traditional drop down interface, whereas Android often has a radio- button list popup, and iOS has a custom scroller covering the

```
<div class="list">

  <label class="item item-input item-select">
    <div class="input-label">
      Lightsaber
    </div>
    <select>
      <option>Blue</option>
      <option selected>Green</option>
      <option>Red</option>
    </select>
  </label>

</div>
```

bottom half of the window.

**11) tabs**

Tabs are a horizontal region of buttons or links that allow for a consistent navigation experience between screens. It can contain any combination of text and icons, and is a popular method for enabling mobile navigation.

```
<div class="tabs">
  <a class="tab-item">
    Home
  </a>
  <a class="tab-item">
    Favorites
  </a>
  <a class="tab-item">
    Settings
  </a>
</div>
```

**12) grid**

Ionic's grid system is different than most because of its use of the CSS Flexible Box Layout Module standard. The advantage here is that the devices that Ionic supports, all support flexbox.

Simply add columns you want in a row, and they'll evenly take up the available space. If you want three columns, add three columns, if you want five columns, add five columns. There's no restriction to a 12 column grid, or having to explicitly state how large each column should be. And to add to the crazy, you can vertically align content within each column.

The `row` class name is used to designate, surprise, a row, and `col` is used for a column. In the demo to the right we chose to have four, then two, columns, but we could have just as easily used 3, 6, 7, 23, etc., it doesn't matter. Point is, add the number of columns your layout requires and don't worry about figuring out the percentages because it figures it out

```
<div class="row">
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
</div>
```

automatically.

## 4. Iconic java script components

1) **Action Sheet-**An Action Sheet is a dialog that displays a set of options. It appears on top of the app's content, and must be manually dismissed by the user before they can resume interaction with the app.

2) **Back Drop-**Backdrops are full screen components that overlay other components. They are useful behind components that transition in on top of other content and can be used to dismiss that component.

3) **Content-**Content component provides an easy to use content area with some useful methods to control the scrollable area. There should only be one content in a single view component.

4) **Model-**Modal controllers programmatically control the modal component. Modals can be created and dismissed from the modal controller.

**5) Popover-**

Things to know when using the popover plug-in:

- Popovers rely on the 3rd party library Popper.js for positioning. You must include popper.min.js before bootstrap.js or use bootstrap.bundle.min.js / bootstrap.bundle.jswhich contains Popper.js in order for popovers to work!
- Popovers require the tooltip plug-in as a dependency.
- If you're building our JavaScript from source, it requires util.js.
- Popovers are opt-in for performance reasons, so you must initialize them yourself.
- Zero-length title and content values will never show a popover.
- Specify container: 'body' to avoid rendering problems in more complex components (like our input groups, button groups, etc).
- Triggering popovers on hidden elements will not work.
- Popovers for .disabled or disabled elements must be triggered on a wrapper element.
- When triggered from anchors that wrap across multiple lines, popovers will be centered between the anchors' overall width. Use .text-now rap on your <a>s to avoid this behavior.
- Popovers must be hidden before their corresponding elements have been removed from the DOM.

Example:-

```
<button type="button" class="btn btn-lg btn-danger" data-toggle="popover"
title="Popover title" data-content="And here's some amazing content. It's very
engaging. Right?">Click to toggle popover</button>
```

6) **Popup-**This service is used for creating a popup window on top of the regular view, which will be used for interaction with the users. There are four types of popup namely – show, confirm, alert and prompt.

7) **Slide Menu-** Side menu is one of the most used Ionic components. The Side menu can be opened by swiping to the left or right or by triggering the button created for that purpose.
8) **Slide Box-**A Slide box contains pages that can be changed by swiping the content screen.

## 5. Platform management in iconic-1

ngCordova comes with over 70 native Cordova plug-in that you can easily add to your Angular Cordova apps. Choose the plug-in you'd like to use from the menu, which will have information on which plug-in you need to install and an example of how to use it in your Angular code.

## 6. Ionic theme

Theme support is baked right into Ionic apps. Changing the theme is as easy as updating the $colors map in your src/theme/variables.scss file: ... The fastest way to change the theme of your Ionic app is to set a new value for primary, since Ionicuses the primary color by default to style most components.

# CHAPTER -4 Interactions with server side PHP

## 1. Database connection to MySQL

PHP 5 and later can work with a MySQL database using:

- MySQLi extension (the "i" stands for improved)
  - PDO (PHP Data Objects)
  - Earlier versions of PHP used the MySQL extension. However, this extension was deprecated in 2012.
  - If you need a short answer, it would be "Whatever you like".

Both MySQLi and PDO have their advantages:

PDO will work on 12 different database systems, whereas MySQLi will only work with MySQL databases.

So, if you have to switch your project to use another database, PDO makes the process easy. You only have to change the connection string and a few queries. With MySQLi, you will need to rewrite the entire code - queries included.

Both are object-oriented, but MySQLi also offers a procedural API.

Both support Prepared Statements. Prepared Statements protect from SQL injection, and are very important for web application security.

## Open a Connection to MySql

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Example (MySQLi Procedural)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);

// Check
connection if
(!$conn) {
   die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

Example (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

try {
   $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
   // set the PDO error mode to exception
   $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
   echo "Connected successfully";
   }
catch(PDOException $e)
   {
   echo "Connection failed: " . $e->getMessage();
   }
?>
```

## Close the Connection

The connection will be closed automatically when the script ends. To close the connection before, use the following:

Example (MySQLi Object-Oriented)

```
$conn->close();
```

Example (MySQLi Procedural)

```
mysqli_close($conn);
```

Example (PDO)

```
$conn = null;
```

## 2. Associative arrays and array handling in PHP

Array_push() - The array_push() function inserts one or more elements to the end of an

array. array_push(array,value1,value2...)

| Parameter | Description |
|---|---|
| array | Required. Specifies an array |
| value1 | Required. Specifies the value to add |
| value2 | Optional. Specifies the value to add |

Example

```
<?php
$a=array("a"=>"red","b"=>"gree
n");
array_push($a,"blue","yellow");
print_r($a);
?>
```

Output

```
Array ( [a] => red [b] => green [0] => blue [1] => yellow )
```

array_pop() - The pop() method removes the last element of an array, and returns that element.

Syntax

```
array.pop()
```

```
<!DOCTYPE html>
<html>
<body>
<p>Click the button to remove the last element from the array.</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML =
fruits; function myFunction() {
  fruits.pop();
  document.getElementById("demo").innerHTML =
  fruits;
}
</script>
</body>
</html>
```

array_search () – This function search an array for a value and returns the key.

## Syntax

array_search(value, array, strict)

| Parameter | Description |
|---|---|
| value | Required. Specifies the value to search for |
| array | Required. Specifies the array to search in |
| strict | Optional. If this parameter is set to TRUE, then this function will search for identical elements in the array. Possible values:<br>• true<br>• false - Default<br>When set to true, the number 5 is not the same as the string 5 |

in_array() – The in_array() function searches an array for a specific value.

## Syntax

in_array(*search*,*array*,*type*)

| **Parameter** | **Description** |
|---|---|
| *search* | Required. Specifies the what to search for |
| *array* | Required. Specifies the array to search |
| *type* | Optional. If this parameter is set to TRUE, the in_array() function searches for the search-string and specific type in the array. |

## **3.**Reading JSON as input

1. file_get_contents("php://input") - Reads entire file into a string

Syntax:-

```
file_get_contents ( string $filename [, bool $use_include_path = FALSE [, resource $context [, int $offset = 0 [, int
$maxlen ]]]] ) : string
```

Explanation:-

This function is similar to file(), except that file_get_contents() returns the file in a string, starting at the specified **offset** up to **maxlen** bytes. On failure, file_get_contents() will return **FALSE**.

2. Json_encode() and json_decode() –

**In PHP, json_encode() is used to convert PHP supported data type into JSON formatted string to be returned as a result of JSON encode operation. This function accepts the following set of arguments.**

- **Data to be encoded.**
- **Options with JSON encode constants to reflect effects on encoding behavior.**

- **Depth limit for performing recursive encoding with nested levels of input.**

**Example: PHP json_encode()**

```
<?php
$input_array = array("zero","one","two");
//returns ["zero","one","two"]
$str_json_format = json_encode($input_array);
print "JSON Formatted String:" .
$str_json_format;
//returns {"0":"zero","1":"one","2":"two"}
$obj_json_format = json_encode($input_array,
JSON_FORCE_OBJECT); print "<br/><br/>JSON Object:" .
$obj_json_format;
//returns [ "zero", "one", "two" ]
$strJsonFormat_with_space = json_encode($input_array, JSON_PRETTY_PRINT);
print "<br/><br/>JSON Formatted String with white space:" . $strJsonFormat_with_space;
?>
```

The second method json_decode() we have listed at the beginning of this article, will be used. This function accepts four arguments as listed below.

- JSON formatted string to be decoded.
- The Boolean value set based on which an associative array will be returned, if it is true.
- depth limit.
- options.

**Example: json_decode()**

```
...

$str_json_array_decoded = json_decode($str_json_format);

print "<br/><br/>Resultant decoded array from JSON

array:<br/>"; print "<PRE>";

print_r($str_json_array_decoded

); print "</PRE>";
```

```
$str_objJson_decoded = json_decode($obj_json_format);

print "<br/><br/>Resultant decoded object data from JSON

object:<br/>"; print "<PRE>";

print_r($str_objJson_decoded

); print "</PRE>";



$str_jsonAry_decoded = json_decode($obj_json_format,true);

print "<br/><br/>Resultant decoded array data from JSON

object:<br/>"; print "<PRE>";

print_r($str_jsonAry_decoded

); print "</PRE>";
```

After that, on executing this code, te following output will be returned to the

browser. Resultant decoded array from JSON array:

```
Arra

y (

   [0] => zero

   [1] => one

   [2] => two

)
```

```
Resultant decoded object data from JSON

object: stdClass Object

(

   [0] => zero

   [1] => one

   [2] => two

)

Resultant decoded array data from JSON object:

Arra

y (

   [0] => zero

   [1] => one

   [2] => two

)
```

# 3. PHP MySql CRUD Application

## What is CRUD

CRUD is an acronym for **C**reate, **R**ead, **U**pdate, and **D**elete. CRUD operations are basic data manipulation for database. We've already learned how to perform create (i.e. insert), read (i.e. select), update and delete operations in previous chapters. In this tutorial we'll create a simple PHP application to perform all these operations on a MySql database table at one place.

## Creating the Database Table

Execute the following SQL query to create a table named *employees* inside your MySQL database. We will use this table for all of our future operations.

CREATE TABLE employees ( id INT NOT NULL PRIMARY KEY AUTO_INCREMENT, name VARCHAR(100) NOT NULL, address VARCHAR(255) NOT NULL, salary INT(10) NOT NULL );

## Creating the Read Page

Now it's time to build the **R**ead functionality of our CRUD application.

Let's create a file named "create.php" and put the following code inside it. It will simply retrieve the records from the *employees* table based the id attribute of the employee.

```
<?php
// Check existence of id parameter before processing further
if(isset($_GET["id"]) && !empty(trim($_GET["id"]))){
    // Include config file
    require_once "config.php";

    // Prepare a select statement
    $sql = "SELECT * FROM employees WHERE id = ?";

    if($stmt = mysqli_prepare($link, $sql)){
        // Bind variables to the prepared statement as parameters mysqli_stmt_bind_param($stmt, "i",
        $param_id);

        // Set parameters
        $param_id = trim($_GET["id"]);

        // Attempt to execute the prepared statement
        if(mysqli_stmt_execute($stmt)){
            $result = mysqli_stmt_get_result($stmt);

            if(mysqli_num_rows($result) == 1){
                /* Fetch result row as an associative array. Since the result set contains only one row, we don't need to use
while loop */
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);

                // Retrieve individual field value
                $name = $row["name"];
                $address = $row["address"];
                $salary = $row["salary"];
            } else{
                // URL doesn't contain valid id parameter. Redirect to error page
                header("location: error.php");
```

```
exit();
```

```
                }

            } else{
                echo "Oops! Something went wrong. Please try again later.";
            }
        }

        // Close statement
        mysqli_stmt_close($stmt);

        // Close connection
        mysqli_close($link);
} else{
        // URL doesn't contain id parameter. Redirect to error page
        header("location: error.php");
        exit();
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <title>View Record</title>
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
        <style type="text/css">
            .wrapper{
                width: 500px;
                margin: 0 auto;
            }
         </style>
</head>
<body>
          <div class="wrapper">
                    <div class="container-fluid">
                        <div class="row">
                                <div class="col-md-12">
                                        <div class="page-header">
                                          <h1>View Record</h1>
                                    </div>
                                    <div class="form-group">
                                            <label>Name</label>
                                            <p class="form-control-static"><?php echo $row["name"]; ?></p>
                                    </div>
                                    <div class="form-group">
                                            <label>Address</label>
                                            <p class="form-control-static"><?php echo $row["address"]; ?></p>
                                    </div>
                                    <div class="form-group">
                                            <label>Salary</label>
                                            <p class="form-control-static"><?php echo $row["salary"]; ?></p>
                                    </div>
                                    <p><a href="index.php" class="btn btn-primary">Back</a></p>
                            </div>
                    </div>
            </div>
        </div>
</body>

</html>
```

## Creating the Update Page

Similarly, we can build the Update functionality of our CRUD application.

Let's create a file named "update.php" and put the following code inside it. It will update the existing records in the employees table based the id attribute of the employee.

```
<?php
// Include config file
require_once "config.php";

// Define variables and initialize with empty values
$name = $address = $salary = "";
$name_err = $address_err = $salary_err = "";

// Processing form data when form is submitted
if(isset($_POST["id"]) && !empty($_POST["id"])){
	// Get hidden input value
	$id = $_POST["id"];

	// Validate name
	$input_name = trim($_POST["name"]);
	if(empty($input_name)){
		$name_err = "Please enter a name.";
	} elseif(!filter_var($input_name, FILTER_VALIDATE_REGEXP, array("options"=>array("regexp"=>"/^[a-zA- Z\s]+$/")))){
		$name_err = "Please enter a valid name.";
	} else{
		$name = $input_name;
	}

	// Validate address address
	$input_address = trim($_POST["address"]);
	if(empty($input_address)){
		$address_err = "Please enter an address.";
	} else{
		$address = $input_address;
	}

	// Validate salary
	$input_salary = trim($_POST["salary"]);
	if(empty($input_salary)){
		$salary_err = "Please enter the salary amount.";
	} elseif(!ctype_digit($input_salary)){
		$salary_err = "Please enter a positive integer value.";
	} else{
		$salary = $input_salary;
	}

	// Check input errors before inserting in database if(empty($name_err) &&
	empty($address_err) && empty($salary_err)){
		// Prepare an update statement
		$sql = "UPDATE employees SET name=?, address=?, salary=? WHERE id=?";

		if($stmt = mysqli_prepare($link, $sql)){
			// Bind variables to the prepared statement as parameters
			mysqli_stmt_bind_param($stmt, "sssi", $param_name, $param_address, $param_salary, $param_id);

			// Set parameters
			$param_name = $name;
			$param_address = $address;
			$param_salary = $salary;
			$param_id = $id;

			// Attempt to execute the prepared statement
			if(mysqli_stmt_execute($stmt)){
```

```
                        exit();
                    } else{
                        echo "Something went wrong. Please try again later.";
                    }
                }

                // Close statement
                mysqli_stmt_close($stmt);
            }

            // Close connection
            mysqli_close($link);
} else{
        // Check existence of id parameter before processing further
        if(isset($_GET["id"]) && !empty(trim($_GET["id"]))){
                // Get URL parameter
                $id =   trim($_GET["id"]);

                // Prepare a select statement
                $sql = "SELECT * FROM employees WHERE id = ?";
                if($stmt = mysqli_prepare($link, $sql)){
                        // Bind variables to the prepared statement as parameters mysqli_stmt_bind_param($stmt, "i", $param_id);

                        // Set parameters
                        $param_id = $id;

                        // Attempt to execute the prepared statement
                        if(mysqli_stmt_execute($stmt)){
                                $result = mysqli_stmt_get_result($stmt);

                                if(mysqli_num_rows($result) == 1){
                                        /* Fetch result row as an associative array. Since the result set contains only one row, we don't need to
use while loop */
                                        $row = mysqli_fetch_array($result, MYSQLI_ASSOC);

                                        // Retrieve individual field value
                                        $name = $row["name"];
                                        $address = $row["address"];
                                        $salary = $row["salary"];
                                } else{
                                        // URL doesn't contain valid id. Redirect to error page
                                        header("location: error.php");
                                        exit();
                                }

                        } else{
                                echo "Oops! Something went wrong. Please try again later.";
                        }
                }

                // Close statement
                mysqli_stmt_close($stmt);

                // Close connection
                mysqli_close($link);
        }   else{
                // URL doesn't contain id parameter. Redirect to error page
                header("location: error.php");
                exit();
        }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
```

```
<title>Update Record</title>
```

```
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
        <style type="text/css">
            .wrapper{
                width: 500px;
                margin: 0 auto;
            }
        </style>
</head>
<body>
     <div class="wrapper">
          <div class="container-fluid">
               <div class="row">
                    <div class="col-md-12">
                         <div class="page-header">
                              <h2>Update Record</h2>
                         </div>
                         <p>Please edit the input values and submit to update the record.</p>
                         <form action="<?php echo htmlspecialchars(basename($_SERVER['REQUEST_URI'])); ?>"
method="post">
                              <div class="form-group <?php echo (!empty($name_err)) ? 'has-error' : ''; ?>">
                                   <label>Name</label>
                                   <input type="text" name="name" class="form-control" value="<?php echo $name;
?>">
                                   <span class="help-block"><?php echo $name_err;?></span>
                              </div>
                              <div class="form-group <?php echo (!empty($address_err)) ? 'has-error' : ''; ?>">
                                   <label>Address</label>
                                   <textarea name="address" class="form-control"><?php echo $address;
?></textarea>
                                   <span class="help-block"><?php echo $address_err;?></span>
                              </div>
                              <div class="form-group <?php echo (!empty($salary_err)) ? 'has-error' : ''; ?>">
                                   <label>Salary</label>
                                   <input type="text" name="salary" class="form-control" value="<?php echo
$salary; ?>">
                                   <span class="help-block"><?php echo $salary_err;?></span>
                              </div>
                              <input type="hidden" name="id" value="<?php echo $id; ?>"/>
                              <input type="submit" class="btn btn-primary" value="Submit">
                              <a href="index.php" class="btn btn-default">Cancel</a>
                         </form>
                    </div>
               </div>
          </div>
     </div>
</body>

</html>
```

## Creating the Delete Page

Finally, we will build the **D**elete functionality of our CRUD application.

Let's create a file named "delete.php" and put the following code inside it. It will delete the existing records from the *employees* table based the id attribute of the employee.

```
<?php
// Process delete operation after confirmation
if(isset($_POST["id"]) && !empty($_POST["id"])){
    // Include config file
    require_once "config.php";

    // Prepare a delete statement
    $sql = "DELETE FROM employees WHERE id = ?";

    if($stmt = mysqli_prepare($link, $sql)){
```

// Bind variables to the prepared statement as parameters mysqli_stmt_bind_param($stmt, "i",
$param_id);

```
                // Set parameters
                $param_id = trim($_POST["id"]);

                // Attempt to execute the prepared statement
                if(mysqli_stmt_execute($stmt)){
                        // Records deleted successfully. Redirect to landing page header("location: index.php");
                        exit();
                } else{
                        echo "Oops! Something went wrong. Please try again later.";
                }
        }

        // Close statement
        mysqli_stmt_close($stmt);

        // Close connection
        mysqli_close($link);
} else{
        // Check existence of id parameter
        if(empty(trim($_GET["id"]))){
                // URL doesn't contain id parameter. Redirect to error page
                header("location: error.php");
                exit();
        }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <title>View Record</title>
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
        <style type="text/css">
                .wrapper{
                        width: 500px;
                        margin: 0 auto;
                }
        </style>
</head>
<body>
        <div class="wrapper">
                <div class="container-fluid">
                    <div class="row">
                        <div class="col-md-12">
                                <div class="page-header">
                                    <h1>Delete Record</h1>
                            </div>
                            <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
                                <div class="alert alert-danger fade in">
                                        <input type="hidden" name="id" value="<?php echo trim($_GET["id"]); ?>"/>
                                        <p>Are you sure you want to delete this record?</p><br>
                                        <p>
                                                <input type="submit" value="Yes" class="btn btn-danger">
                                                <a href="index.php" class="btn btn-default">No</a>
                                        </p
                                        >
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
        </div>
</body>

</html>
```

**Creating the Error Page**

At the end, let's create one more file "error.php". This page will be displayed if request is invalid i.e. if id parameter is missing from the URL query string or it is not valid.

```
<!DOCTYPE html>
<html lang="en">
 <head> <meta charset="UTF-8">
 <title>Error</title>
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
<style type="text/css"> .wrapper{ width: 750px; margin: 0 auto; } </style>
 </head>
 <body>
 <div class="wrapper">
 <div class="container-fluid">
 <div class="row">
 <div class="col-md-12">
 <div class="page-header">
<h1>Invalid Request</h1>
 </div>
 <div class="alert alert-danger fade in">
 <p>Sorry, you've made an invalid request. Please <a href="index.php" class="alert-link">go back</a> and try
again.
</p> </div> </div> </div> </div> </div> </body> </html>
```

## 4. Introduction to Nodejs

**What is Node.js?**

- Node.js is an open source server environment
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js uses JavaScript on the server

**Why Node.js?**

A common task for a web server can be to open a file on the server and return the content to the client. Here is how PHP or ASP handles a file request:

1. Sends the task to the computer's file system.
2. Waits while the file system opens and reads the file.
3. Returns the content to the client.
4. Ready to handle the next request.

Here is how Node.js handles a file request:

1. Sends the task to the computer's file system.
2. Ready to handle the next request.
3. When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient.

### What Can Node.js Do?

- Node.js can generate dynamic page content
- Node.js can create, open, read, write, delete, and close files on the server
- Node.js can collect form data
- Node.js can add, delete, modify data in your database

### What is a Node.js File?

- Node.js files contain tasks that will be executed on certain events
- A typical event is someone trying to access a port on the server
- Node.js files must be initiated on the server before having any effect
- Node.js files have extension ".js"

## 5. Architecture of Nodejs

### myfirst.js

```
var http = require('http');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type':
  'text/html'}); res.end('Hello World!');
}).listen(8080);
```

Save the file on your computer: C:\Users\*Your Name*\myfirst.js
The code tells the computer to write **"Hello World!"** if anyone (e.g. a web browser) tries to access your computer on port **8080**.

## 6. Step by step installation of Nodejs

### Prerequisites

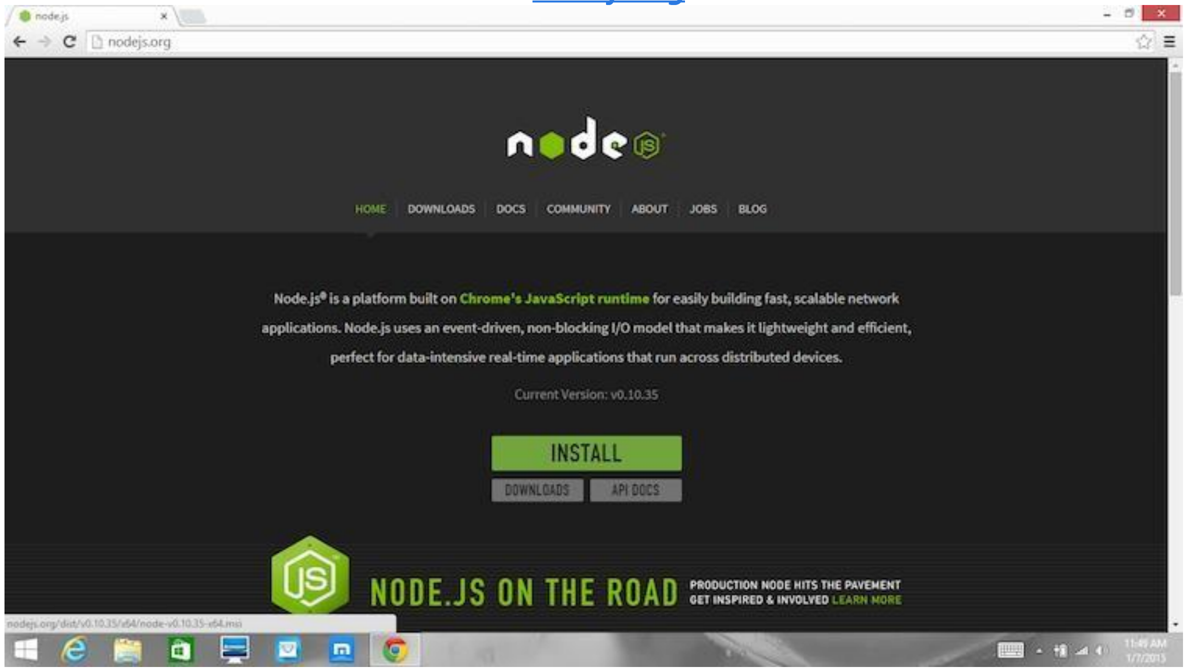
- **You should have some familiarity with an application that lets you issue command line instructions.** For example, the Windows Command Prompt, PowerShell, Cygwin, or the Git shell (which you get when you install Github for Windows).

### Installation Overview

Installing Node.js® and NPM is pretty straightforward using the installer package available from the Node.js web site.

### Installation Steps

1. **Download the Windows installer from Nodejs.org.**

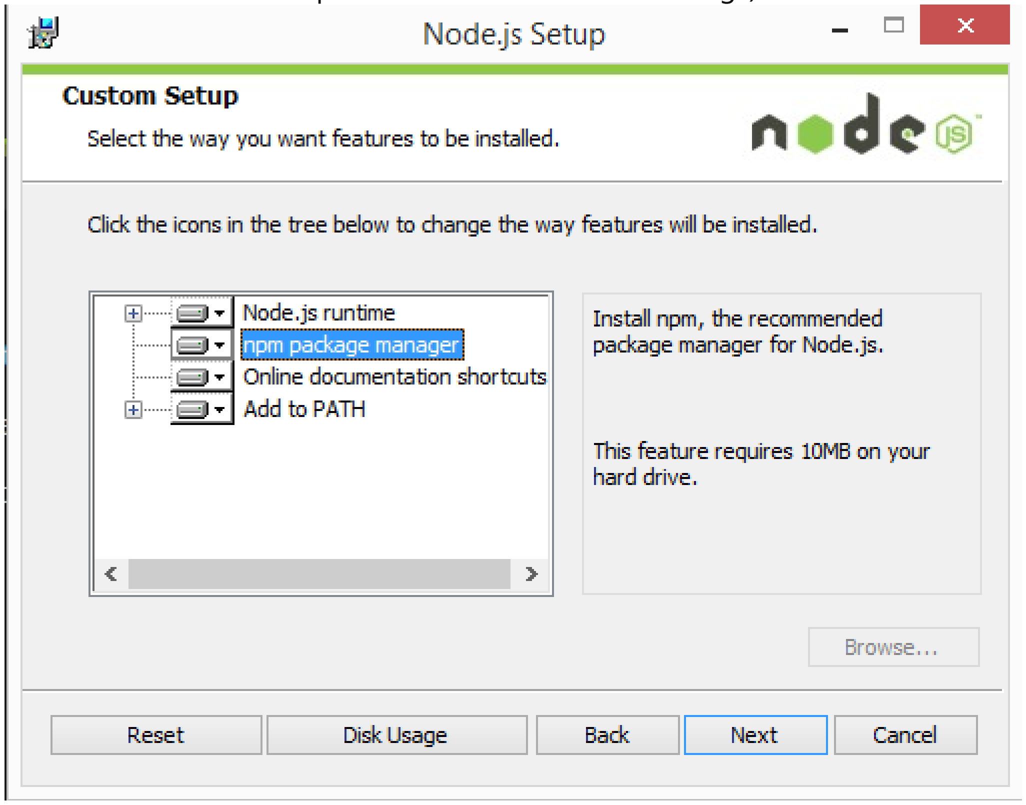


Node.js® is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Current Version: v0.10.35

2. **Run the installer** (the .msi file you downloaded in the previous step.)

3. **Follow the prompts in the installer** (Accept the license agreement, click the NEXT button a bunch of times and accept the default installation settings).


Node.js Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

- Node.js runtime
- npm package manager
- Online documentation shortcuts
- Add to PATH

Install npm, the recommended package manager for Node.js.

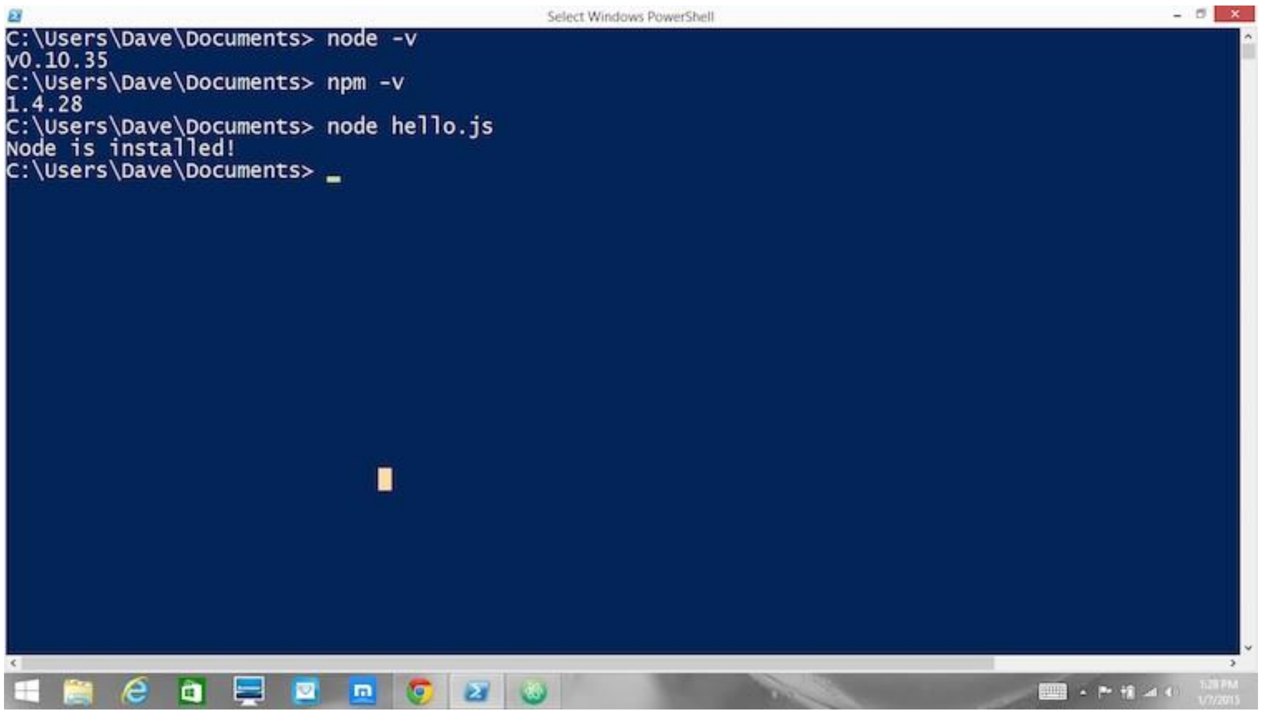
This feature requires 10MB on your hard drive.

4. **Restart your computer.** You won’t be able to run Node.js until you restart your computer.

## Test it!

Make sure you have Node and NPM installed by running simple commands to see what version of each is installed:

- **Test Node.** To see if Node is installed, open the Windows Command Prompt, Powershell or a similar command line tool, and type `node -v` . This should print the version number so you’ll see something like this `v0.10.35` .
- **Test NPM.** To see if NPM is installed, `npm -v` in Terminal. This should print the version number type so you’ll see something like this `1.4.28`
- **Create a test file and run it.** A simple way to test that node.js works is to create a simple JavaScript file: name it hello.js, and just add the code `console.log('Node is installed!');` . To run the code simply open your command line program, navigate to the folder where you save the file and type `node hello.js`. This will start Node.js and run the code in the `hello.js` file. You should see the output `Node is installed!`



```
C:\Users\Dave\Documents> node -v
v0.10.35
C:\Users\Dave\Documents> npm -v
1.4.28
C:\Users\Dave\Documents> node hello.js
Node is installed!
C:\Users\Dave\Documents> _
```

## 7. Introduction to express-Nodejs

Express.js, or simply Express, is a web application framework for Node.js, released as free and open- source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

**History**

Express.js was founded by TJ Holowaychuk. The first release, according to Express.js's GitHub repository, was on the 22nd of May, 2010. Version 0.12.0

In June 2014, rights to manage the project were acquired by StrongLoop. StrongLoop was acquired by IBM in September 2015; in January 2016, IBM announced that it would place Express.js under the stewardship of the Node.js Foundation incubator.

## 8. Create a server and listen to port in Nodejs

**myfirst.js**

```
var http = require('http');

http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type':
  'text/html'}); res.end('Hello World!');
}).listen(8080);
```

Save the file on your computer: C:\Users\*Your Name*\myfirst.js
The code tells the computer to write **"Hello World!"** if anyone (e.g. a web browser) tries to access your computer on port **8080**.

# CHAPTER – 5 Accessing Native Services using Iconic and Application Signing and Development

## 1. IONIC –CORDOVA INTEGRATION

**Manage Cordova plugins**

$ionic cordova plugin [<action>] [<plugin>] [options]

**Examples**

ionic cordova plugin

ionic cordova plugin add cordova-plugin-inappbrowser@latestcordova-plugin-

inappbrowser@latest ionic cordova plugin add phonegap-plugin-push --variable

SENDER_ID=XXXXX ionic cordova plugin rm cordova-plugin-camera

**Inputs**

| Action | |
|---|---|
| **Description** | addor removea plugin; lsor saveall project plugins |

| Plugin | |
|---|---|
| **Description** | The name of the plugin (corresponds to addand remove) |

## 2. Ionic –camera

Take a photo or capture video

Requires the Cordova plugin: cordova-plugin-camera

**Installation**

1. Install the Cordova and Ionic Native plugins:
2. $ ionic cordova plugin add cordova-plugin-camera
3. $ npm install --save @ionic-native/camera@4
4. Add this plugin to your app's module

**Usage**

```
import { Camera, CameraOptions } from '@ionic-

native/camera'; constructor(private camera: Camera) { }

...


const options: CameraOptions
  = { quality: 100,
  destinationType:
  this.camera.DestinationType.FILE_URI,
  encodingType: this.camera.EncodingType.JPEG,
  mediaType: this.camera.MediaType.PICTURE
}

this.camera.getPicture(options).then((imageData) => {
 // imageData is either a base64 encoded string or a file URI
 // If it's base64 (DATA_URL):
 let base64Image = 'data:image/jpeg;base64,' + imageData;
}, (err) => {
 // Handle error
});
```

**CameraOptions**

| Param | Type | Details |
|---|---|---|
| quality | number | Picture quality in range 0-100. Default is 50<br><br>*(optional)* |
| destinationType | number | Choose the format of the return value. Defined in Camera.DestinationType. Default is FILE_URI. DATA_URL : 0, Return image as base64-encoded string (DATA_URL can be very memory intensive and cause app crashes or out of memory errors. Use FILE_URI or NATIVE_URI if possible), FILE_URI : 1, Return image file URI, NATIVE_URI : 2 Return image native URI (e.g., assets-library:// on iOS or content:// on Android)<br><br>*(optional)* |

| | | |
|---|---|---|
| sourceType | number | Set the source of the picture. Defined in Camera.PictureSourceType. Default is CAMERA. PHOTOLIBRARY : 0, CAMERA : 1, SAVEDPHOTOALBUM : 2<br><br>*(optional)* |

| Param | Type | Details |
|---|---|---|
| allowEdit | boolean | Allow simple editing of image before selection. *(optional)* |
| encodingType | number | Choose the returned image file's encoding. Defined in Camera.EncodingType. Default is JPEG JPEG : 0 Return JPEG encoded image PNG : 1 Return PNG encoded image *(optional)* |
| targetWidth | number | Width in pixels to scale image. Must be used with targetHeight. Aspect ratio remains constant. *(optional)* |
| targetHeight | number | Height in pixels to scale image. Must be used with targetWidth. Aspect ratio remains constant. *(optional)* |
| mediaType | number | Set the type of media to select from. Only works when PictureSourceType is PHOTOLIBRARY or SAVEDPHOTOALBUM. Defined in Camera.MediaType PICTURE: 0 allow selection of still pictures only. DEFAULT. Will return format specified via DestinationType VIDEO: 1 allow selection of video only, WILL ALWAYS RETURN FILE_URI ALLMEDIA : 2 allow selection from all media types *(optional)* |
| correctOrientation | boolean | Rotate the image to correct for the orientation of the device during capture. *(optional)* |
| saveToPhotoAlbum | boolean | Save the image to the photo album on the device after capture. *(optional)* |
| cameraDirection | number | Choose the camera to use (front- or back-facing). Defined in Camera.Direction. Default is BACK. BACK: 0 FRONT: 1 *(optional)* |

| | | |
|---|---|---|
| popoverOptions | CameraPopoverOptions | iOS-only options that specify popover location in iPad. Defined in CameraPopoverOptions.<br><br>*(optional)* |

**CameraPopoverOptions**

| Param | Type | Details |
|---|---|---|
| x | number | |
| y | number | |
| width | number | |
| height | number | |
| arrowDir | number | Direction the arrow on the popover should point. Defined in Camera.PopoverArrowDirection Matches iOS UIPopoverArrowDirection constants. ARROW_UP : 1, ARROW_DOWN : 2, ARROW_LEFT : 4, ARROW_RIGHT : 8, ARROW_ANY : 15 |

## 3. Ionic-native audio

**Installation**

```
$ionic cordova plugin add cordova-plugin-nativeaudio

 $npm install @ionic-native/native-audio
```

**Usage**

```
import { NativeAudio } from '@ionic-native/native-

audio/ngx'; constructor(private nativeAudio:

NativeAudio) { }

...

this.nativeAudio.preloadSimple('uniqueId1', 'path/to/file.mp3').then(onSuccess, onError);
this.nativeAudio.preloadComplex('uniqueId2', 'path/to/file2.mp3', 1, 1, 0).then(onSuccess,
onError);

this.nativeAudio.play('uniqueId1').then(onSuccess, onError);

// can optionally pass a callback to be called when the file is done playing
this.nativeAudio.play('uniqueId1', () => console.log('uniqueId1 is done
playing'));

this.nativeAudio.loop('uniqueId2').then(onSuccess, onError);

this.nativeAudio.setVolumeForComplexAsset('uniqueId2',

0.6).then(onSuccess,onError);

this.nativeAudio.stop('uniqueId1').then(onSuccess,onError);
this.nativeAudio.unload('uniqueId1').then(onSuccess,onError);
```

## 4. Ionic-media

**Installation**

ionic cordova plugin add cordova-plugin-

media npm install @ionic-native/media

**Usage**

```
import { Media, MediaObject } from '@ionic-native/media/ngx';

constructor(private media: Media) { }
...
// Create a Media instance. Expects path to file or url as argument
// We can optionally pass a second argument to track the status of the

media const file: MediaObject = this.media.create('file.mp3');

// to listen to plugin events:

file.onStatusUpdate.subscribe(status => console.log(status)); // fires when file status

changes file.onSuccess.subscribe(() => console.log('Action is successful'));

file.onError.subscribe(error => console.log('Error!', error));

// play the
file file.play();

// pause the
file file.pause();

// get current playback position
file.getCurrentPosition().then((position)
=> { console.log(position);
});

// get file duration
let duration =
file.getDuration();
console.log(duration);

// skip to 10 seconds (expects int value in
ms) file.seekTo(10000);

// stop playing the
file file.stop();
```

```
// release the native audio resource
// Platform Quirks:
// iOS simply create a new instance and the old one will be overwritten
// Android you must call release() to destroy instances of media when you
are done file.release();

// Recording to a file
const file: MediaObject = this.media.create('path/to/file.mp3');

file.startRecord()
;
file.stopRecord()
;
```

## 5. Ionic-inapp browser

**Installation**

Install the Cordova and Ionic Native plugins:

```
$ ionic cordova plugin add cordova-plugin-inappbrowser
$ npm install --save @ionic-native/in-app-browser@4
```

**Usage**

```
import { InAppBrowser } from '@ionic-native/in-app-browser';

constructor(private iab: InAppBrowser) { }
...
const browser =
this.iab.create('https://ionicframework.com/');
browser.executeScript(...);
browser.insertCSS(...);
browser.on('loadstop').subscribe(event
=> {
  browser.insertCSS({ code: "body{color: red;" });
});

browser.close(
);
```

## 6. Introduction to git and its commands

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM (Software Configuration Manager) tools like Subversion, CVS, Perforce, and Clear Case with features like cheap local branching, convenient staging areas, and multiple workflows.

## Git commands

| | |
|---|---|
| Git init Initialize a local Git repository | |
| git clone ssh://git@github.com/[username]/[repository-name].git | |

## Create a local copy of a remote repository

| | |
|---|---|
| git status | Check status |
| git add [file-name.txt] | Add a file to the staging area |
| git add –A | Add all new and changed files to the staging area |

git commit -m "[commit message]"
Commit changes

git rm -r [file-name.txt] Remove a file (or folder)

| | |
|---|---|
| git branch | List branches (the asterisk denotes the current branch) |
| git branch –a | List all branches (local and remote) |
| git branch [branch name] | Create a new branch |
| git branch -d [branch name] | Delete a branch |
| git push origin --delete [branchName] | Delete a remote branch |
| git checkout -b [branch name] | Create a new branch and switch to it |
| git checkout -b [branch name] origin/[branch name] | Clone a remote branch and switch to it |
| git checkout [branch name] | Switch to a branch |
| git checkout - | Switch to the branch last checked out |
| git checkout -- [file-name.txt] | Discard changes to a file |
| git merge [branch name] | Merge a branch into the active branch |
| git merge [source branch] [target branch] | Merge a branch into a target branch |
| git stash | Stash changes in a dirty working directory |
| git stash clear | Remove all stashed entries |
| git push origin [branch name] | Push a branch to your remote repository |
| git push -u origin [branch name] | Push changes to remote repository (and remember the branch) |

# 7. Git Branches

Branch operation allows creating another line of development. We can use this operation to fork off the development process into two different directions. For example, we released a product for 6.0 version and we might want to create a branch so that the development of 7.0 features can be kept separate from 6.0 bug fixes.

**Create Branch**

Tom creates a new branch using the git branch <branch name> command. We can create a new branch from an existing one. We can use a specific commit or tag as the starting point. If any specific commit ID is not provided, then the branch will be created with HEAD as its starting point.

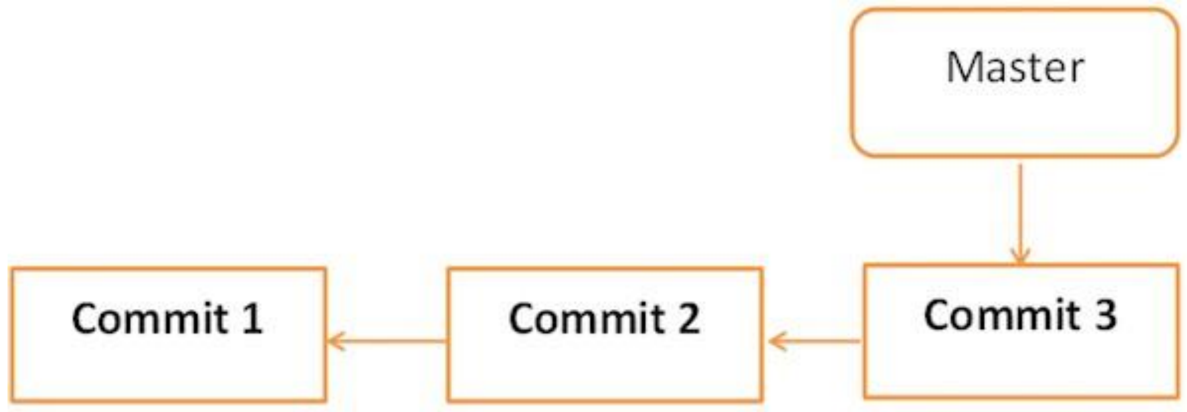
```
[jignesh@CentOS src]$ git branch

new_branch [jignesh@CentOS src]$ git

branch
* master
new_branc
h
```

A new branch is created; Tom used the git branch command to list the available branches. Git shows an asterisk mark before currently checked out branch.

The pictorial representation of create branch operation is shown below –



Before create branch command

# Android

## 1. Signing application (keystore)

Android requires that all apps be digitally signed with a certificate before they can be installed. Android uses this certificate to identify the author of an app, and the certificate does not need to be signed by a certificate authority. Android apps often use self-signed certificates. The app developer holds the certificate's private key.

### Signing Your App in Android Studio

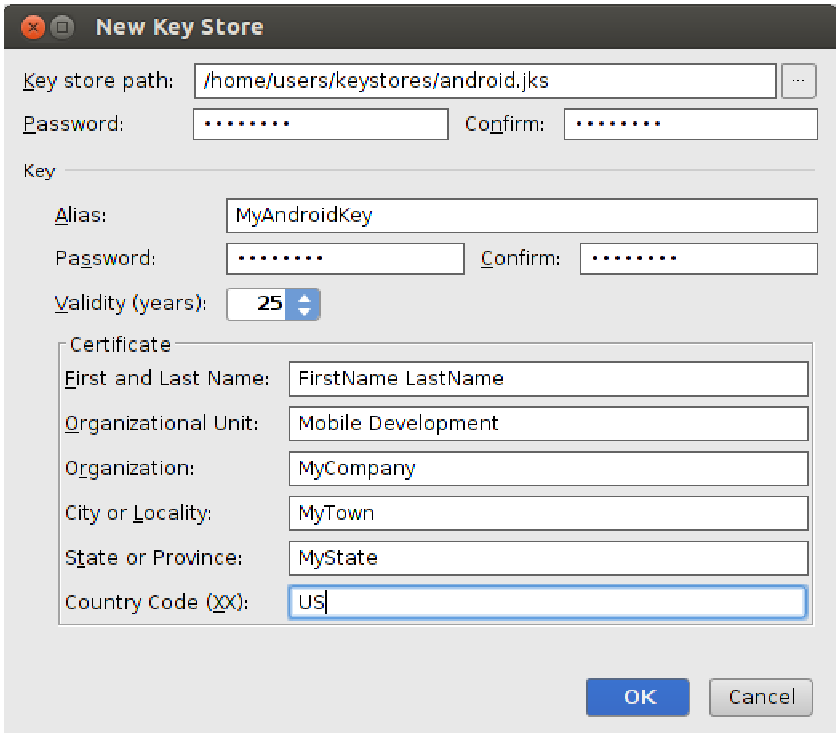
To sign your app in release mode in Android Studio, follow these steps:

1. On the menu bar, click **Build** > **Generate Signed APK**.
2. On the *Generate Signed APK Wizard* window, click **Create new** to create a new keystore.

   If you already have a keystore, go to step 4.

3. On the *New Key Store* window, provide the required information as shown in figure 1.

   Your key should be valid for at least 25 years, so you can sign app updates with the same key through the lifespan of your app.



New Key Store

| | |
|---|---|
| Key store path: | /home/users/keystores/android.jks |
| Password: | •••••••• |
| Confirm: | •••••••• |

Key

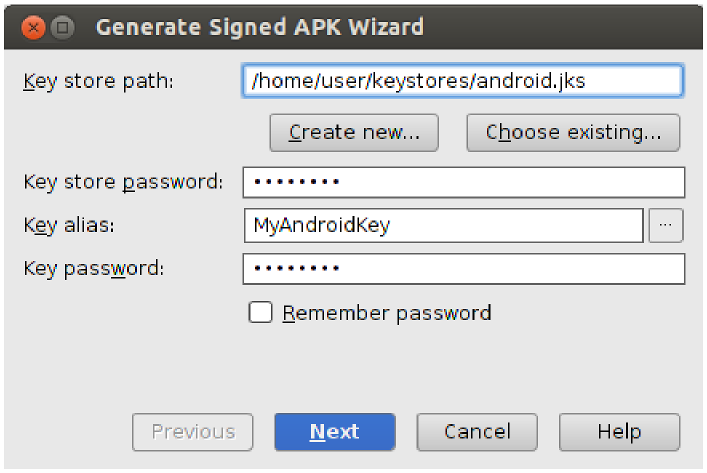
| | |
|---|---|
| Alias: | MyAndroidKey |
| Password: | •••••••• |
| Confirm: | •••••••• |
| Validity (years): | 25 |

Certificate

| | |
|---|---|
| First and Last Name: | FirstName LastName |
| Organizational Unit: | Mobile Development |
| Organization: | MyCompany |
| City or Locality: | MyTown |
| State or Province: | MyState |
| Country Code (XX): | US |

**Figure 1**. Create a new keystore in Android Studio.

4. On the *Generate Signed APK Wizard* window, select a keystore, a private key, and enter the passwords for both. Then click **Next**.



Generate Signed APK Wizard

Key store path: /home/user/keystores/android.jks

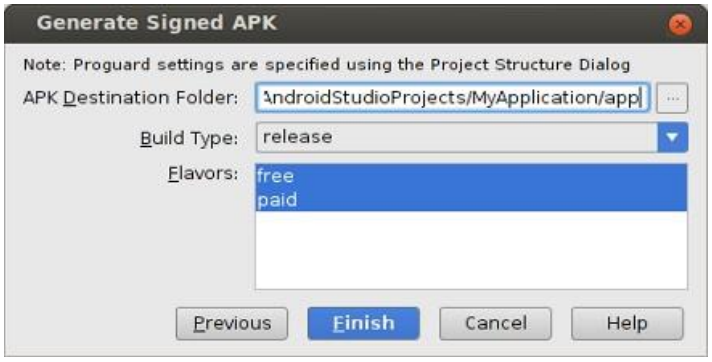
Key store password: ••••••••

Key alias: MyAndroidKey

Key password: ••••••••

**Figure 2**. Select a private key in Android Studio.

5. On the next window, select a destination for the signed APK and click **Finish**.



Generate Signed APK

Note: Proguard settings are specified using the Project Structure Dialog

APK Destination Folder: \AndroidStudioProjects/MyApplication/app

Build Type: release

Flavors: free, paid

**Figure 3**. Generate a signed APK in Android Studio.

o Publishing application on play store

1. Create an account

Firstly, to publish your app on google play store, you need to have account on google. You may have already personal email account with them but for your app it's better to separate one for manage your app(s) easily. You would have to pay a registration fees of 25 USD, using Google payments, While registering your publisher account. After that, a verification mail would be sent to you and then you sign in to your developer console, where all action would take place.

2. **Familiarize yourself with Developer Console**

Developer console is starting point and main dashboard for publishing tools and operations. Before you go ahead, familiarize yourself with list of merchant countries and developer countries. You need to review list of merchant countries if you want to sell apps or in app purchases or have subscriptions and list of developer countries will tell you about locations where distribution to Google play users is supported. Apart from this, also looks the Google Play's Terms and conditions.

3. Fill in the necessary account details

After this, you have to provide your complete account details like you have to provide your developer name, the name which would be displayed in Google Play Store. You will have to wait for anything between just a little and 48 hours, after filling the details for Google play developer registration to be processed.

4. **Link your merchant account**

If you have a paid app or subscription or in app purchases, then you have to inter link your google payments merchant account and developer profile. It can be used for tax and financial identification as well as monthly payout from sale.

5. Upload your app

When you are sign in your Google Play developer console, click on "Add new application" from "All Applications tab". After this select correct "Default Language" from drop down menu and then type "title" of your app which would be appear on Google Play store. Then, select "Upload APK" button to view on new page which would be your homepage for app. Time taken to upload file depend on your app size. App will be in drafts until or unless you publish it.

6. **Alpha and beta testing of app**

It is essential to test it with sample of end users to get feedback of app before launch your app even Google play take care of this as well. In section of your app "APK" of developer console, you will find option related to "Alpha Testing" and "Beta Testing". After you uploaded your app's "APK" file you can use these options for receive URL that can be shared with testers. By using this link, Testers can download app beta or alpha version. Your testers can not provide reviews and feedback on app page. Now you use this feedback to make relevant changes and optimise your app before publishing in app store.

7. Provide details for store listing

After uploading "APK" file go to "Store listing" tab. Over there you need to add details of app like "Full description" and "Short description". Along with this add categorization, contact details, link of promo video if you have, add screenshots and other necessary details to app. After complete mandatory fields click on "Save" button. You can update your store listing anytime.

8. Add pricing and distribution details

Now move on next tab, which is "Pricing and Distribution" and select it is "Paid" or "Free" app. You also select distribution countries and check boxes stating your app comply with content guidelines. If your app is a game one, then you can put in limelight using "Google Play for Game" option. Now save changes and move on next step.

9. Publishing the application

When all three tabs "Pricing and Distribution", "Store Listing" and "APK" have been filled then appear a green check mark next to them you are all ready to publish app in Google Play. After then click on "Publish this app" button under "Ready to Publish" drop down menu at top right corner of Developer console. A confirmation bar would show up stating your app would appear shortly in Google Play Store.

10. Device filtering option

There are series of extra option that might not seem to be important to publish the app but they can prevent app from negative feedback. There is also option to manually filter non compatible devices or problematic so make the most use of it to filter out any negativities and stay on the top.

# IOS SECTION

## 1. Build and publication application in App store

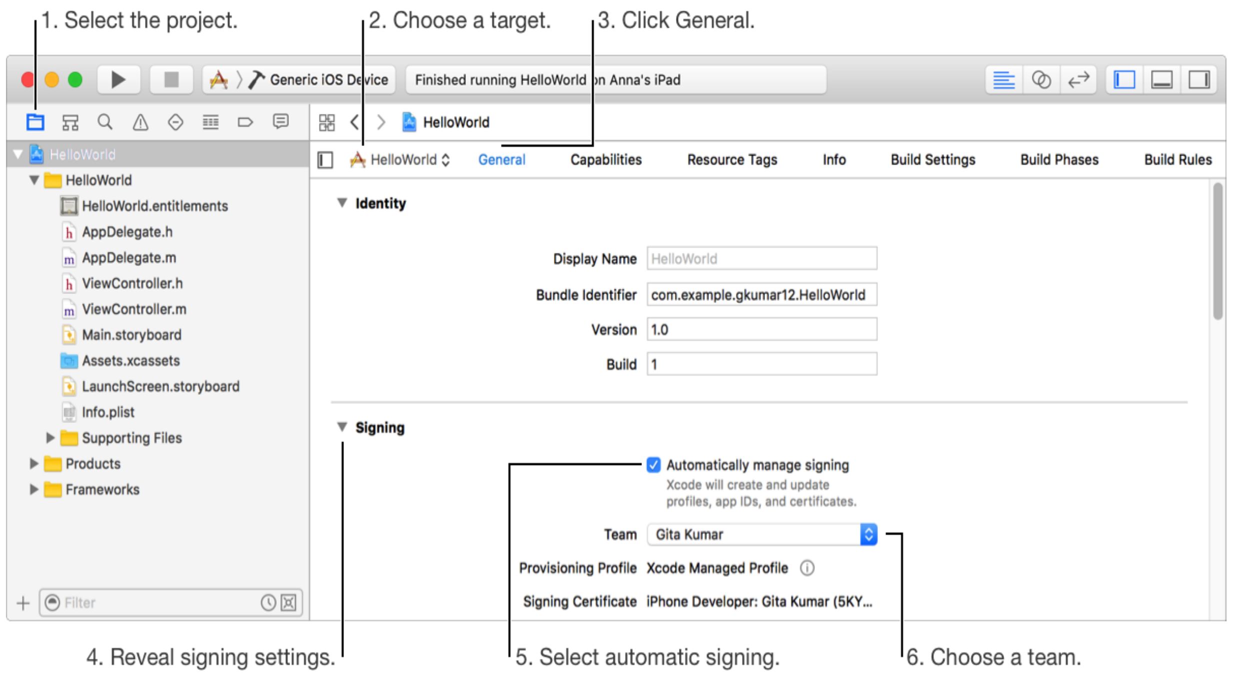
**1.** Code Signing: Create an iOS distribution provisioning profile and distribution certificate

The development provisioning profile and development certificate that you've been using are only for specific devices. In order to distribute your app to beta testers or to users through the App Store, you'll need a separate distribution provisioning profile and distribution certificate.

The easiest way to do this is through Xcode. If automatic signing is enabled, Xcode will create and manage certificates, signing identities, and handle device registration for you. If automatic signing is already enabled or if you don't need help with signing, you can skip to step two.

(In some cases, you might prefer manual signing. Here's an in-depth tutorial for how to manually sign your app. Keep in mind that all targets in a bundle should use the same signing method.)

- First, add your Developer Program account to Xcode if you haven't already. From the top menu, select "Xcode", then choose "Preferences".
- Click on "Accounts". In the bottom left corner of the window, press the "+" sign, then "Add Apple ID…".
- Enter the Apple ID and password you use for the Apple Developer Program, then click "Sign In".
- Next, enable automatic signing. From the Project Editor, choose a target and select "General".
- Scroll down to the "Signing" section and click on the triangle icon to expand the settings.
- Click on the box to "Automatically manage signing". Select your team.



Finished running HelloWorld on Anna's iPad

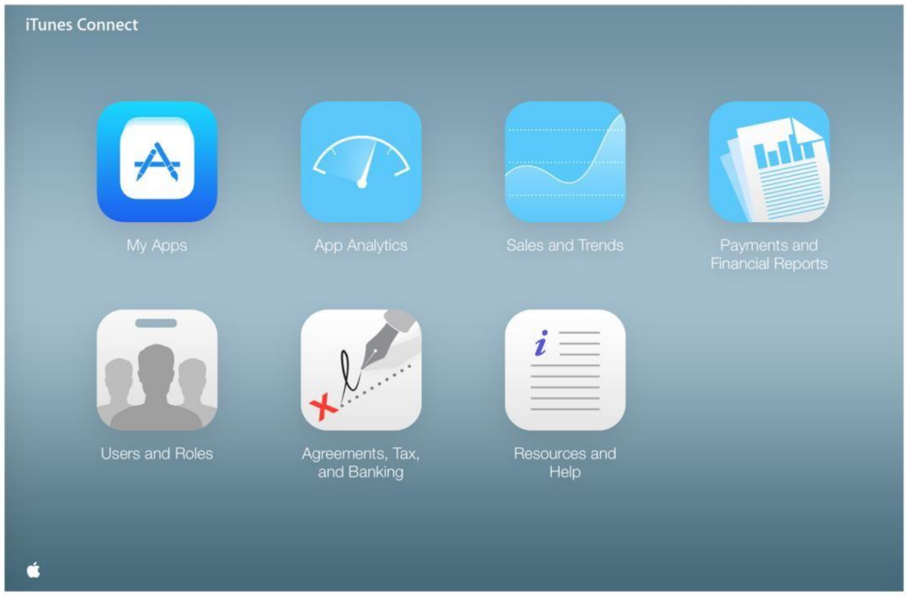
Display Name HelloWorld
Bundle Identifier com.example.gkumar12.HelloWorld
Version 1.0
Build 1

✓ Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.

Team Gita Kumar
Provisioning Profile Xcode Managed Profile
Signing Certificate iPhone Developer: Gita Kumar (5KY...

When you connect a new device to your Mac, Xcode will automatically detect and register it to your team provisioning profile. Note that in order to launch your app on a device, the device needs to be registered on your team provisioning profile.

**2.** Create an iTunes Connect record for your app



**Get an iTunes Connect account by:**

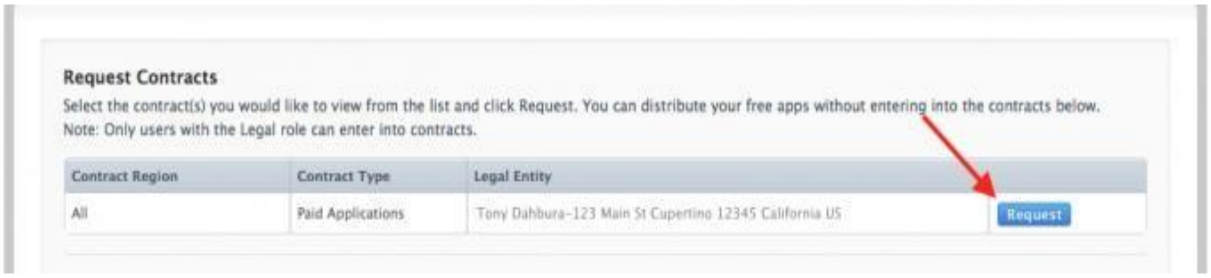
- Creating your own iTunes Connect organization and being the team agent. Sign in with the Apple ID you used to enroll in the Apple Developer Program
- Or being invited by an existing organization as a user with an Admin, Technical, or App Manager role. Read more details about iTunes Connect user accounts here.

For paid apps

If you're submitting a paid app, you'll need to sign a contract that covers terms of payment. If your app is free, you can skip ahead.

Click on "**Agreements, Tax, and Banking**" on the iTunes Connect

dashboard. Click on "Request" under "Request Contracts".



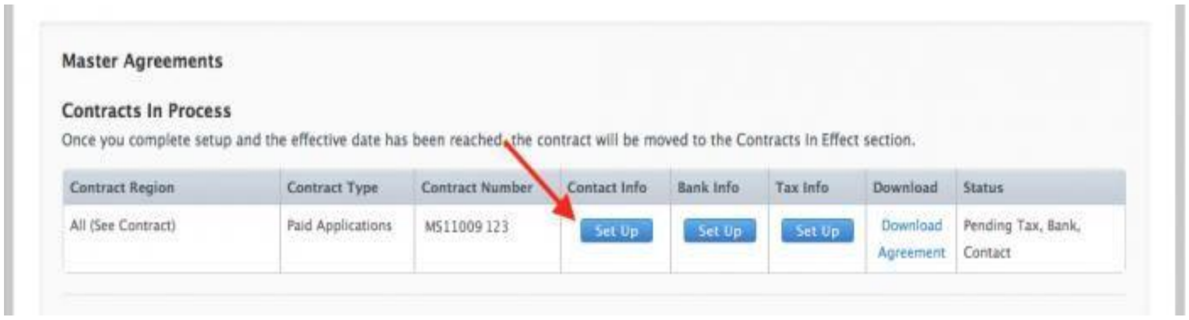
**Request Contracts**

Select the contract(s) you would like to view from the list and click Request. You can distribute your free apps without entering into the contracts below.
Note: Only users with the Legal role can enter into contracts.

| Contract Region | Contract Type | Legal Entity | |
|---|---|---|---|
| All | Paid Applications | Tony Dahbura-123 Main St Cupertino 12345 California US | Request |

Review the agreement that appears, check the box to agree to the terms, and click on

"Submit". Under "Contracts In Process" click "Set Up" in the "**Contact Info**" column.



**Master Agreements**

**Contracts In Process**

Once you complete setup and the effective date has been reached, the contract will be moved to the Contracts In Effect section.

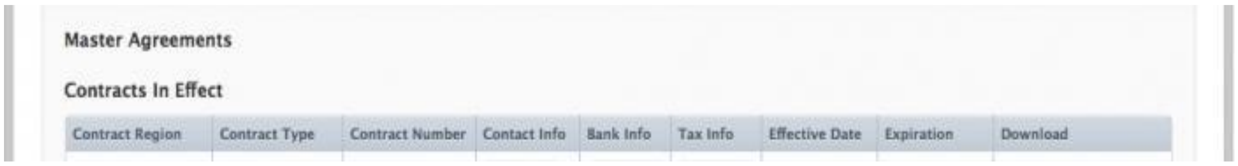
| Contract Region | Contract Type | Contract Number | Contact Info | Bank Info | Tax Info | Download | Status |
|---|---|---|---|---|---|---|---|
| All (See Contract) | Paid Applications | MS11009 123 | Set Up | Set Up | Set Up | Download Agreement | Pending Tax, Bank, Contact |

In the window that appears, click on "Add New Contact" and enter your information.

Back under "Contracts In Process" in the "**Bank Info**" column, click "Set Up" then "Add Bank Account" and follow the directions to save your account info.

In the "**Tax Info**" column, click "Set Up". A U.S. Tax Form is *mandatory*, so click "Set Up" and fill out the required information. Set up any other country tax forms necessary.

After you've completed the above, the contract's status will now say "Processing". After Apple has verified the info you provided, which will take about an hour, the contract will now appear under "Contracts In Effect".



Master Agreements

Contracts In Effect

| Contract Region | Contract Type | Contract Number | Contact Info | Bank Info | Tax Info | Effective Date | Expiration | Download |
|---|---|---|---|---|---|---|---|---|

## Add a new app

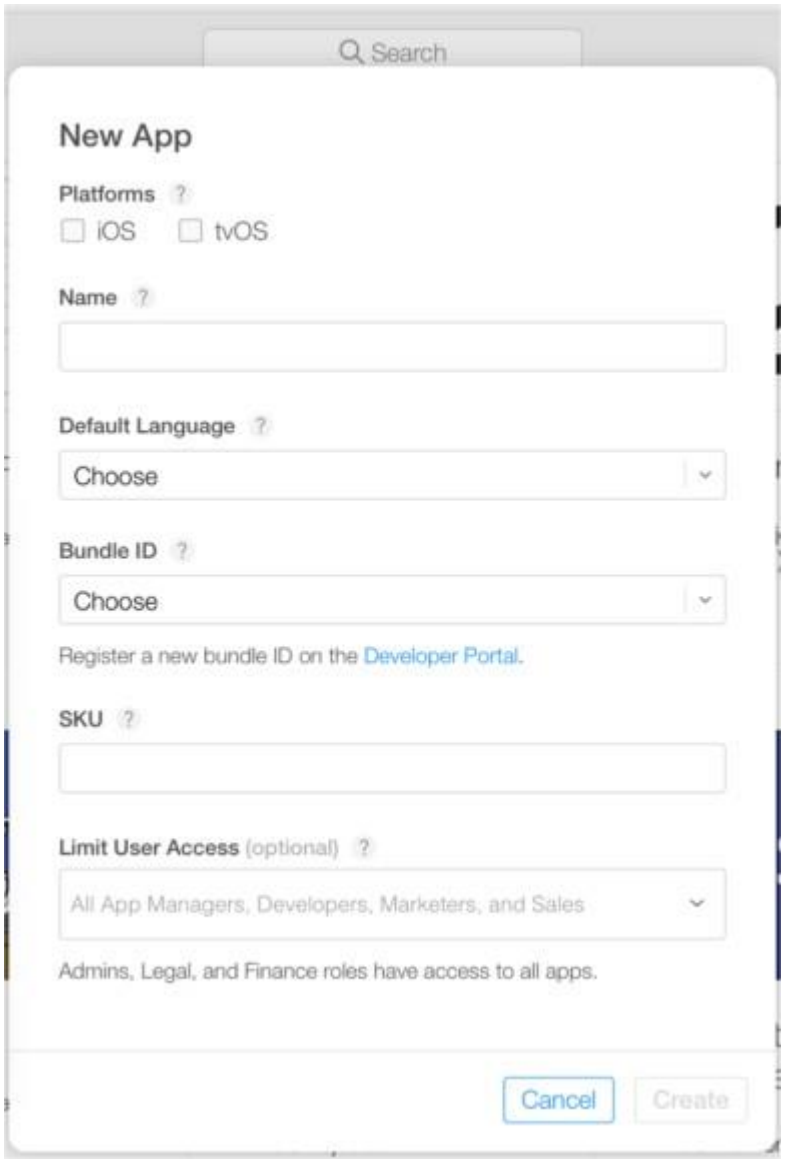
In the iTunes Connect dashboard, select "**My Apps**".

Click on the "+" sign in the upper left-hand corner, then "New App".

To create a new iTunes Connect record, you'll need these details: **platform**, **app name**, **default language**, **bundle ID**, and **SKU**. You can't really change these details later, so be sure of what you enter.

- Use keywords in your app name to optimize for discovery.
- The bundle ID must be an exact match of the bundle identifier in your Xcode project Info.plist file (in the target's General > Identity section).
- The SKU is not visible to users and is up to you to set. It can be an identifier you use in your company or something else that is meaningful for you. Acceptable characters include letters, numbers, hyphens, periods, and underscores, and it must begin with a letter or number.

If applicable, you can also set user access at this step.


Search
New App
Platforms
iOS
tvOS
Name
Default Language
Choose
Bundle ID
Choose
Register a new bundle ID on the Developer Portal.
SKU
Limit User Access (optional)
All App Managers, Developers, Marketers, and Sales
Admins, Legal, and Finance roles have access to all apps.
Cancel
Create

**3.** Archive and upload your app using Xcode

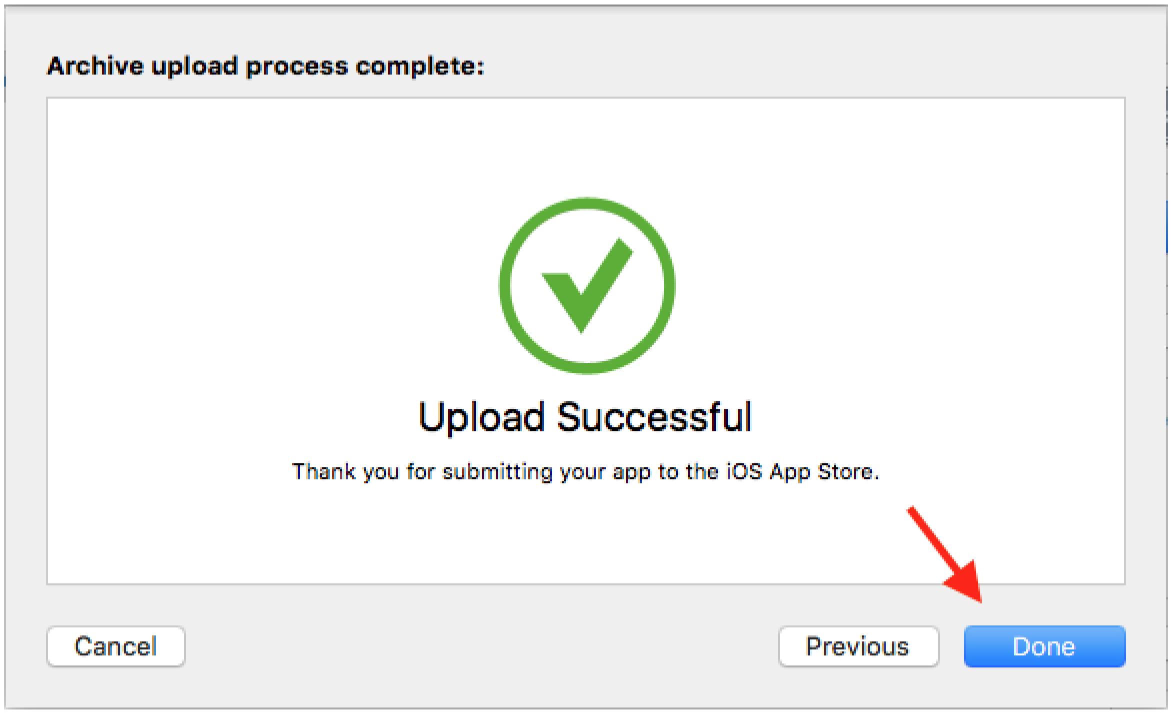
Before you can submit your app for review through iTunes Connect, you need to upload the build through Xcode.

- In Xcode, select “Generic iOS Device” as the deployment target.
- Choose “Product” from the top menu and click on “Archive”.
- The Xcode Organizer will launch, displaying any archives you’ve created in the past.

- Make sure the current build is selected and click on “Upload to App Store” in the right-hand panel.
- Select your credentials and click “Choose”.
- In the next window that appears, click on “Upload” in the bottom right-hand

corner. An “Upload Successful” message will appear when the upload has completed.

Click “Done”.



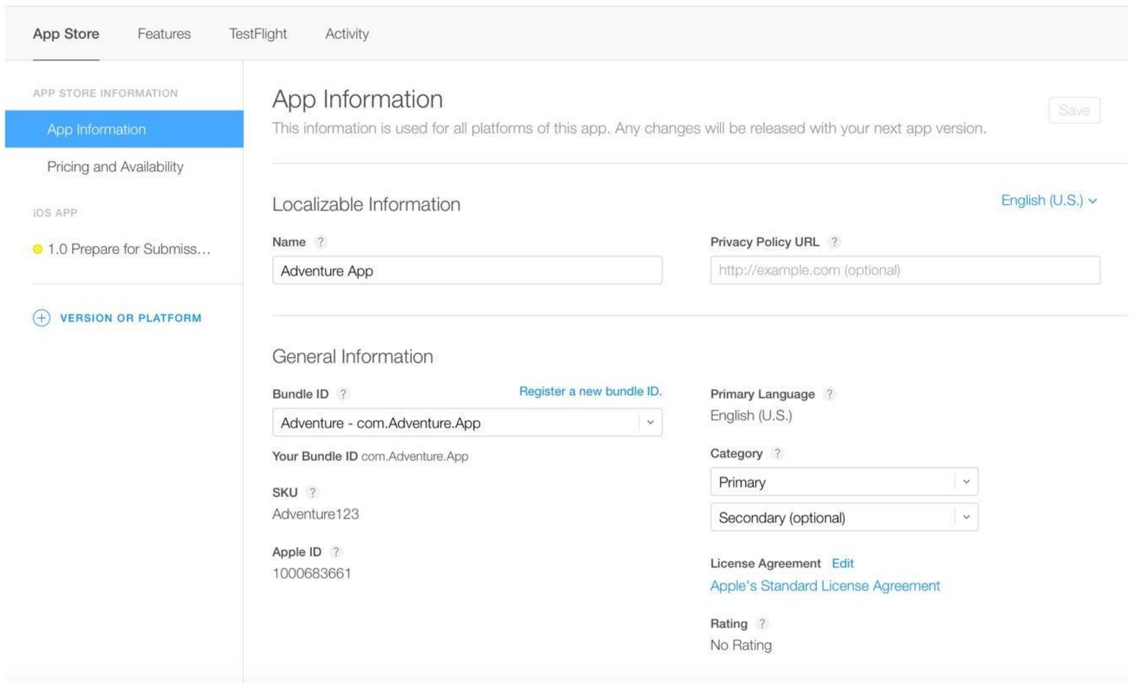
**Archive upload process complete:**

Upload Successful

Thank you for submitting your app to the iOS App Store.

**4.** Configure your app’s metadata and further details in its iTunes Connect record

Under the “App Store” tab in iTunes Connect, in the “App Information” page you can add **additional languages**, **categories**, and your app’s **Privacy Policy URL**.


App Store | Features | TestFlight | Activity

APP STORE INFORMATION

App Information

Pricing and Availability

iOS APP

1.0 Prepare for Submiss…

VERSION OR PLATFORM

App Information

This information is used for all platforms of this app. Any changes will be released with your next app version.

Save

Localizable Information

English (U.S.)

Name: Adventure App

Privacy Policy URL: http://example.com (optional)

General Information

Bundle ID: Adventure - com.Adventure.App

Register a new bundle ID.

Your Bundle ID com.Adventure.App

SKU: Adventure123

Apple ID: 1000683661

Primary Language: English (U.S.)

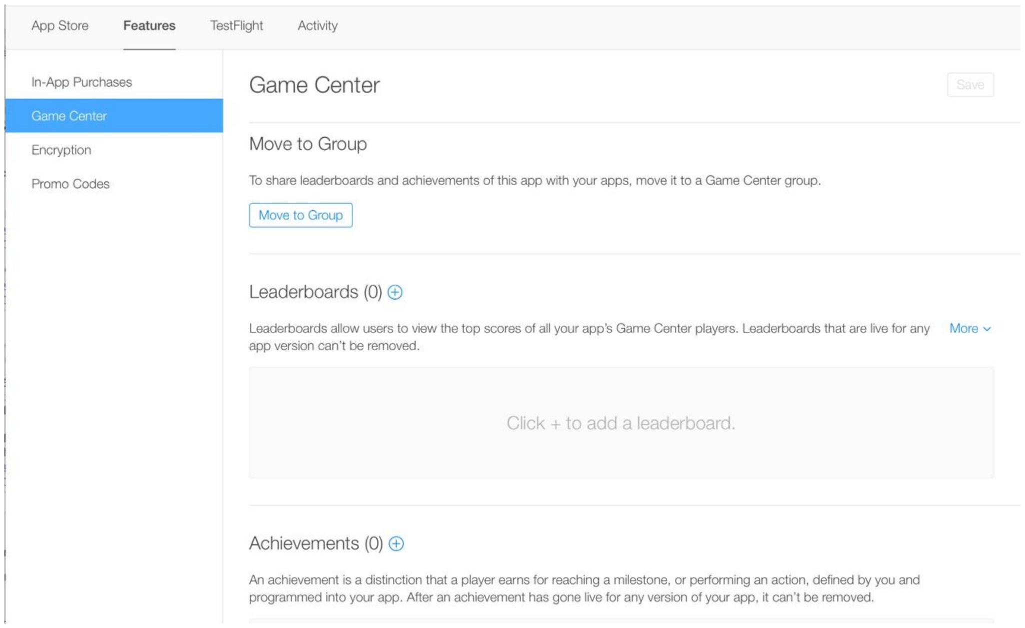
Category: Primary / Secondary (optional)

License Agreement Edit: Apple's Standard License Agreement

Rating: No Rating

Set your app as **free** or select its **price tier** on the "Pricing and Availability" page.

Under the "Features" tab, you can add configurations to any **App Store technologies** in your app, like Game Center and in-app purchases.



Game Center

Move to Group

To share leaderboards and achievements of this app with your apps, move it to a Game Center group.

Leaderboards (0)

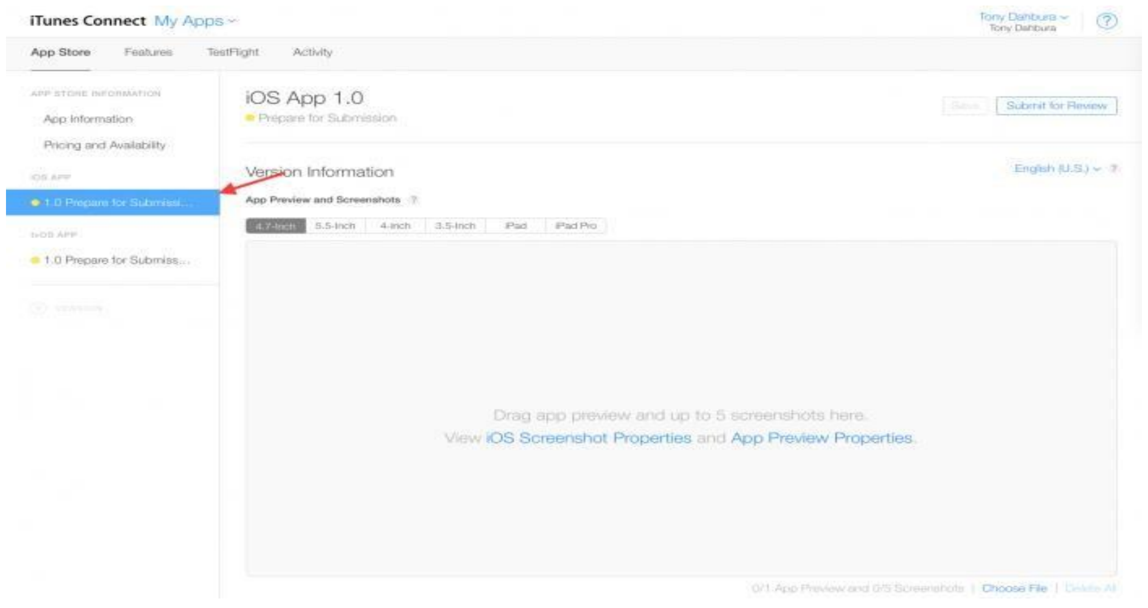
Leaderboards allow users to view the top scores of all your app's Game Center players. Leaderboards that are live for any app version can't be removed.

Click + to add a leaderboard.

Achievements (0)

An achievement is a distinction that a player earns for reaching a milestone, or performing an action, defined by you and programmed into your app. After an achievement has gone live for any version of your app, it can't be removed.

At this stage, your app is marked with a yellow dot and the status “Prepare for Submission” in the left- hand panel under “App Store”. Select the build you want to configure. This is where you will add the information for your **product page** on the App Store.



iOS App 1.0

Prepare for Submission

Version Information

App Preview and Screenshots

Drag app preview and up to 5 screenshots here.
View iOS Screenshot Properties and App Preview Properties.

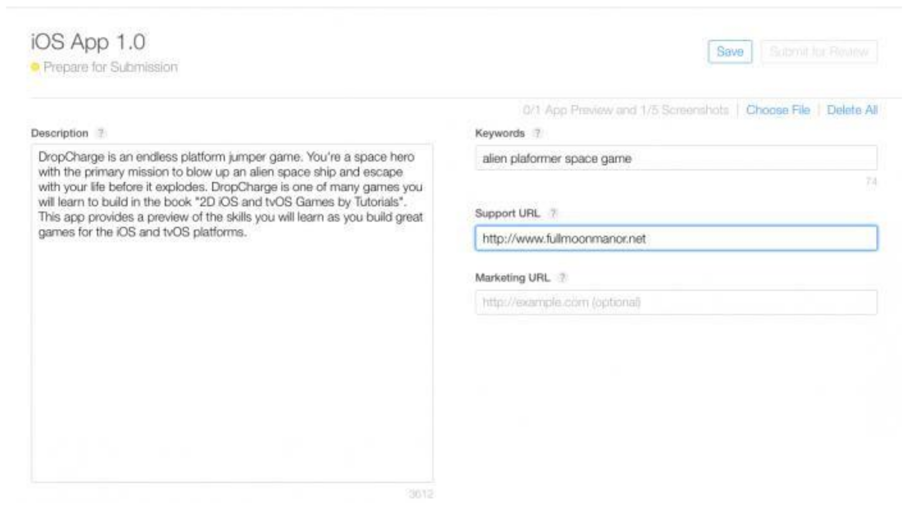
0/1 App Preview and 0/5 Screenshots

Upload your app’s **screenshots** (in JPEG or PNG format and without status bars). You can upload a set of screenshots for one device and use them for all the other sizes.

Click on “Save” in the upper right-hand corner of the window after your screenshots have finished uploading.

Scroll down and enter your app’s **description**, **keywords**, **support URL**, and **marketing URL**.

- Your app’s description and keywords are critical. Make sure you optimize them for discovery.
- The support URL can be as simple as a landing page with a contact form.
- The marketing URL can be your app’s website and is optional.



Description

DropCharge is an endless platform jumper game. You're a space hero with the primary mission to blow up an alien space ship and escape with your life before it explodes. DropCharge is one of many games you will learn to build in the book "2D iOS and tvOS Games by Tutorials". This app provides a preview of the skills you will learn as you build great games for the iOS and tvOS platforms.

Keywords: alien plaformer space game

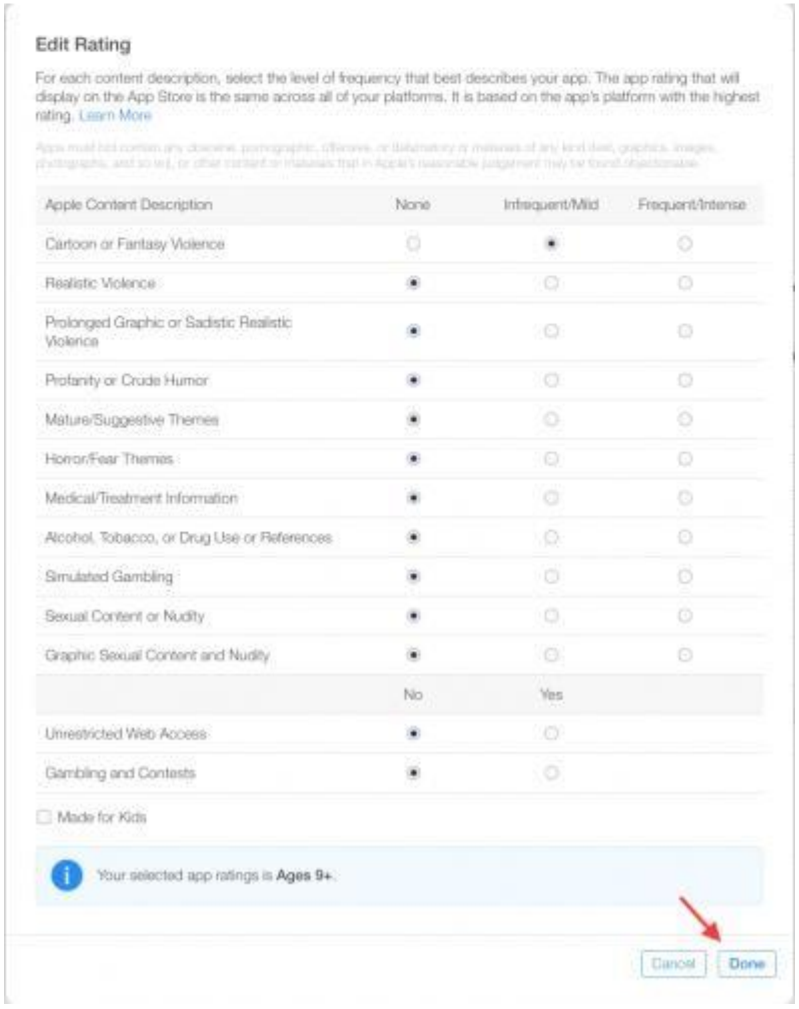
Support URL: http://www.fullmoonmanor.net

Marketing URL: http://example.com (optional)

In the “General App Information” section below, upload your app’s **icon**, enter its **version number**, and
**copyright** and **contact** information.

- Your app’s icon should be 1024px x 1024px.
- The version number should exactly match the one in Xcode.
- The copyright info typically looks like this: “Copyright (c) 2017, Instabug, Inc.”
- The contact info here is what will be displayed to users.

Click on “Edit” next to “**Rating**” and select the applicable options for your app. Be honest — your app can be rejected during review if it doesn’t match its rating.



Edit Rating

For each content description, select the level of frequency that best describes your app. The app rating that will display on the App Store is the same across all of your platforms. It is based on the app's platform with the highest rating. Learn More

Apps must not contain any obscene, pornographic, offensive, or defamatory or materials of any kind (text, graphics, images, photographs, and so on), or other content or materials that in Apple's reasonable judgement may be found objectionable.

| Apple Content Description | None | Infrequent/Mild | Frequent/Intense |
|---|---|---|---|
| Cartoon or Fantasy Violence | ○ | ● | ○ |
| Realistic Violence | ● | ○ | ○ |
| Prolonged Graphic or Sadistic Realistic Violence | ● | ○ | ○ |
| Profanity or Crude Humor | ● | ○ | ○ |
| Mature/Suggestive Themes | ● | ○ | ○ |
| Horror/Fear Themes | ● | ○ | ○ |
| Medical/Treatment Information | ● | ○ | ○ |
| Alcohol, Tobacco, or Drug Use or References | ● | ○ | ○ |
| Simulated Gambling | ● | ○ | ○ |
| Sexual Content or Nudity | ● | ○ | ○ |
| Graphic Sexual Content and Nudity | ● | ○ | ○ |
| | No | Yes | |
| Unrestricted Web Access | ● | ○ | |
| Gambling and Contests | ● | ○ | |

☐ Made for Kids

Your selected app ratings is **Ages 9+**.

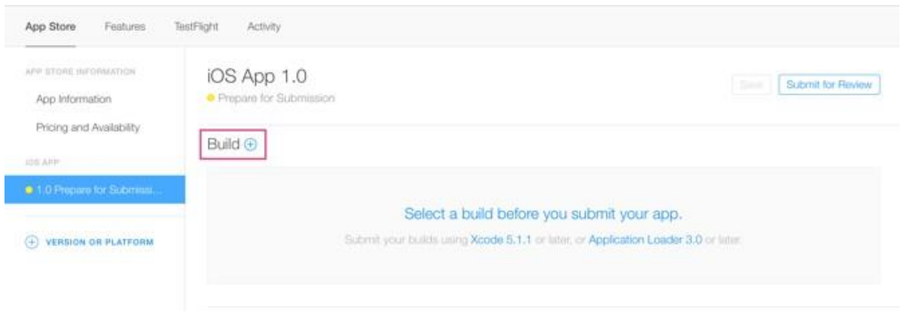
Under the “App Review Information” section, enter your **contact** info, any **notes** you have for the reviewer, and set the **version release date.**

- The contact information here is for the reviewer in case they need to reach you directly.
- Notes for the reviewer can include information about specific hardware they might need to use or user account information they might need for access.
- For first releases, you should typically leave the version release date as automatic.

In the top right-hand corner, click “Save”. Now you’re almost ready to “Submit for Review”.

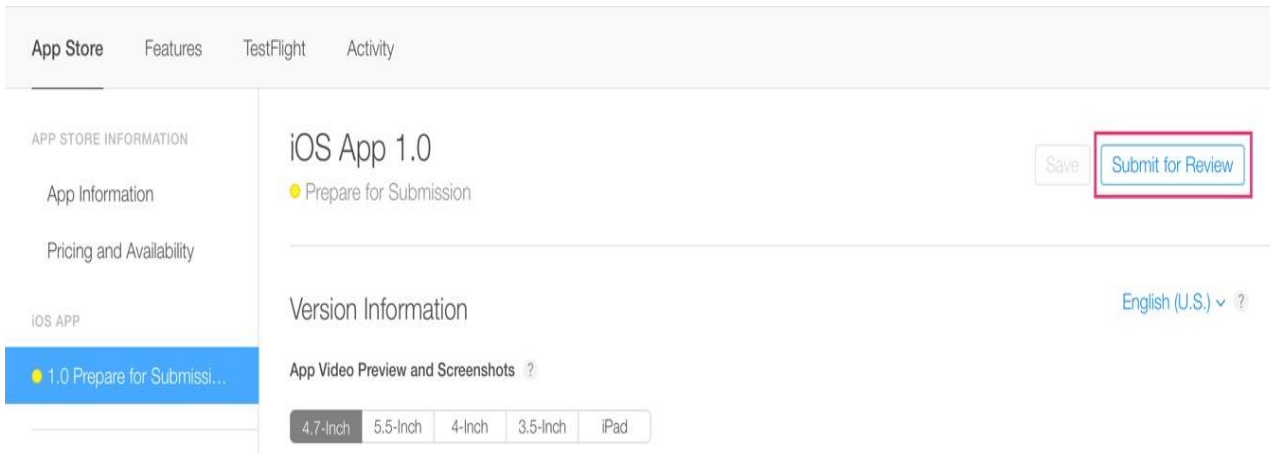
**5.** Submit your app for review

Scroll to the “Build” section in your app’s iTunes Connect record.

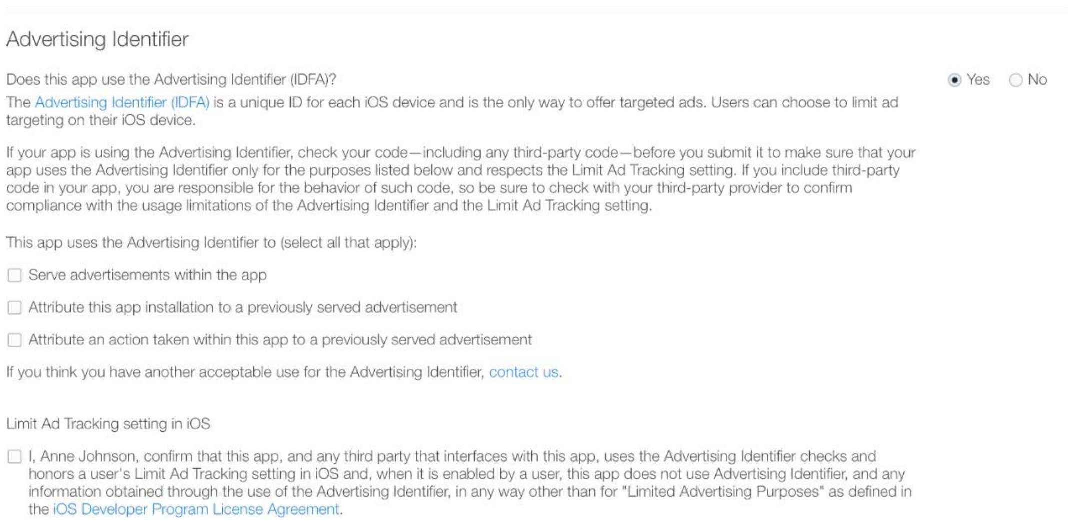


Click on “Select a build before you submit your app.”

Choose the build that you uploaded through Xcode. Click “Done” in the bottom right-hand corner, then “Save” in the top-right hand corner, then “Submit for Review”.



Finally, answer the **Export Compliance**, **Content Rights**, and **Advertising Identifier** questions and click “Submit”.



Advertising Identifier

Does this app use the Advertising Identifier (IDFA)? [✓ Yes] No

The Advertising Identifier (IDFA) is a unique ID for each iOS device and is the only way to offer targeted ads. Users can choose to limit ad targeting on their iOS device.

If your app is using the Advertising Identifier, check your code—including any third-party code—before you submit it to make sure that your app uses the Advertising Identifier only for the purposes listed below and respects the Limit Ad Tracking setting. If you include third-party code in your app, you are responsible for the behavior of such code, so be sure to check with your third-party provider to confirm compliance with the usage limitations of the Advertising Identifier and the Limit Ad Tracking setting.

This app uses the Advertising Identifier to (select all that apply):

- ☐ Serve advertisements within the app
- ☐ Attribute this app installation to a previously served advertisement
- ☐ Attribute an action taken within this app to a previously served advertisement

If you think you have another acceptable use for the Advertising Identifier, contact us.

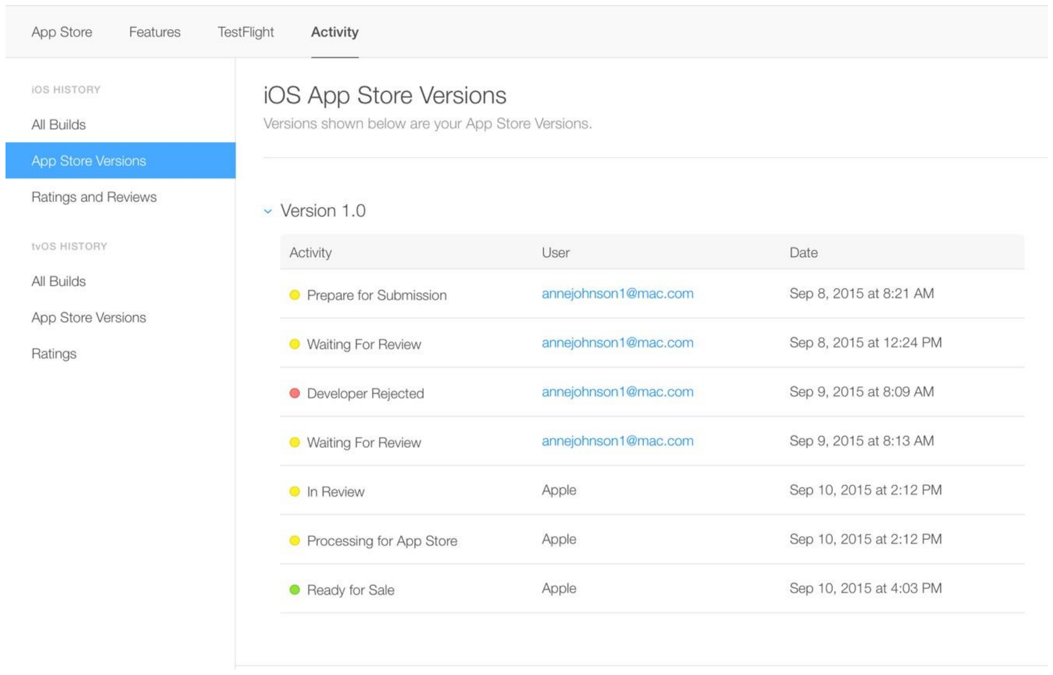
Limit Ad Tracking setting in iOS

- ☐ I, Anne Johnson, confirm that this app, and any third party that interfaces with this app, uses the Advertising Identifier checks and honors a user's Limit Ad Tracking setting in iOS and, when it is enabled by a user, this app does not use Advertising Identifier, and any information obtained through the use of the Advertising Identifier, in any way other than for "Limited Advertising Purposes" as defined in the iOS Developer Program License Agreement.

Your app's status is now "Waiting For Review". ?

**6.** Check on the status of your app

In iTunes Connect, select "Activity" in the top horizontal menu, then "App Store Versions" in the left- hand panel.



iOS App Store Versions

Versions shown below are your App Store Versions.

Version 1.0

| Activity | User | Date |
|---|---|---|
| ● Prepare for Submission | annejohnson1@mac.com | Sep 8, 2015 at 8:21 AM |
| ● Waiting For Review | annejohnson1@mac.com | Sep 8, 2015 at 12:24 PM |
| ● Developer Rejected | annejohnson1@mac.com | Sep 9, 2015 at 8:09 AM |
| ● Waiting For Review | annejohnson1@mac.com | Sep 9, 2015 at 8:13 AM |
| ● In Review | Apple | Sep 10, 2015 at 2:12 PM |
| ● Processing for App Store | Apple | Sep 10, 2015 at 2:12 PM |
| ● Ready for Sale | Apple | Sep 10, 2015 at 4:03 PM |

## How long does it take to get App Store approval?

In most cases, it takes about one to three days to receive approval, and it can take up to 24 hours for your app to appear in the App Store after approval. Check current average app store review times here.

You'll receive e-mail notifications at each stage. Read more about each status here.

If you're on a tight timeline and need to align your release with a specific event or if you need to release a new version with an urgent bug fix, you can request an expedited review.

## If your app is rejected

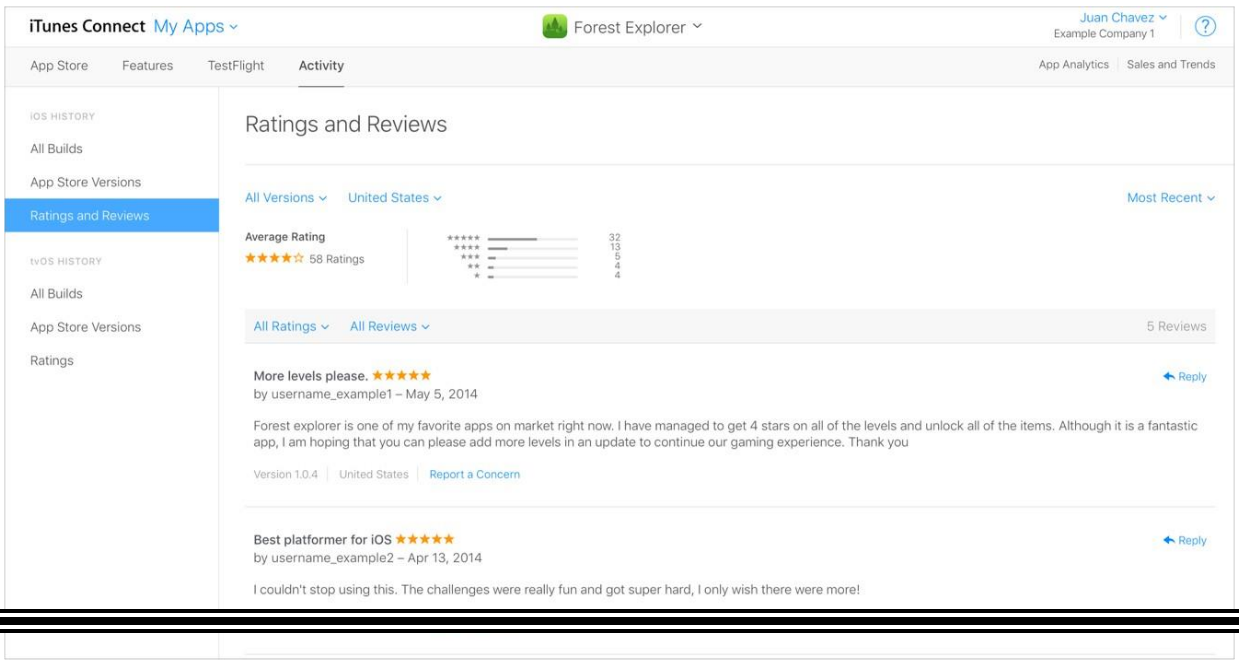
You'll have to make the necessary fixes before you can submit your app again for review. Use the Resolution Center in iTunes Connect to communicate with Apple about any questions you may have. You can also escalate the issue and submit an appeal if you believe your app was wrongly rejected.

One of the most common reasons for rejections from the Apple App Store is performance. Make sure

that your app is complete and that you've tested it thoroughly and fixed all bugs. Using a bug reporting tool while beta testing will help reduce your app's chances of being rejected due to performance issues.

## If your app is approved

Congratulations! You're now on the App Store. You can view downloads, sales, ratings, and reviews directly in iTunes Connect.



Ratings and Reviews

Average Rating ★★★★☆ 58 Ratings

| | |
|---|---|
| ★★★★★ | 32 |
| ★★★★ | 13 |
| ★★★ | 5 |
| ★★ | 4 |
| ★ | 4 |

More levels please. ★★★★★
by username_example1 – May 5, 2014

Forest explorer is one of my favorite apps on market right now. I have managed to get 4 stars on all of the levels and unlock all of the items. Although it is a fantastic app, I am hoping that you can please add more levels in an update to continue our gaming experience. Thank you

Version 1.0.4 | United States

Best platformer for iOS ★★★★★
by username_example2 – Apr 13, 2014

I couldn't stop using this. The challenges were really fun and got super hard, I only wish there were more!