i18n Ionic 1.6 Documentation

Quick start Guide - Installation and user support

Created: July 20, 2015 Updated: May 05, 2016

By: Stavros Kounis, about.me/stavros.kounis

Thank you for purchasing my product. If you have any question that are beyond of the scope of this help file, please feel free to email via my user page contact form. Thank you!

Table of contents:

Prerequisites

PhoneGap and Ionic

Tools

Run for the first time

Download and extract

Install libraries

Post installation

Windows Users

Plugins

Run a local development server

Run in the emulator

How Internationalisation works

Introduction

Configuration

Configuring a new language

Language files and Translations

Language tags / codes

Naming language codes

Alternative locale creation

A working example

Home

Plain Text

Date/Time

Controls

Default language

Available languages

Add a new language/locale

Support

References / Links:

Thank you

Prerequisites

PhoneGap and Ionic

This is a <u>PhoneGap</u> and <u>Ionic</u> based application, so the <u>PhoneGap</u> and <u>Ionic</u> should be installed in your computer. Since **i18n Ionic**, targets iPhone and Android mobile devices, your environment should be properly configured and the corresponding SDK should be installed. If not you will be still able to run the application into a Browser.

Please check the "Install PhoneGap" and "Getting Started with Ionic" sections in the official PhoneGap and Ionic sites respectively.

Ensure first that **NodeJS** is installed in your computer

Tools

This project is based on the popular "<u>lonic Framework Generator</u>" that boosts the overall development process by integrating a couple of very popular automation tools like <u>Grunt</u> and <u>Bower</u>.

Install these tools by following the instructions in their corresponding web pages:

- 1. Install Bower
- 2. Getting started with Grunt Install the CLI
- 3. Getting started with Yeoman

Finally install the yeoman generator via:

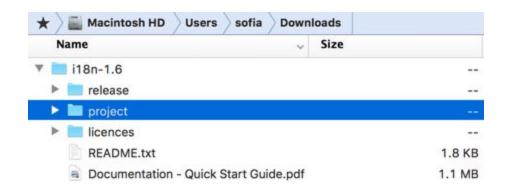
\$ npm install -g generator-ionic

Run for the first time

In the screen captures that follow, we will demonstrate the process of preparing your environment and running the project for the first time.

Download and extract

Download the provided .zip file and extract it, you will see something similar to what is shown in the next screen:



The highlighted folder is used for all the instructions in this document.

Install libraries

Open a terminal window and navigate to the directory where the "i18n" folder is located.

Install NodeJS dependences:

\$ npm install

Post installation

There is a post installation process under which required Cordova plugins and Javascript dependencies are installed. To simplify this process two scripts are already prepared for both platforms: Linux/MacOS and Windows

Linux/MacOX

Install all the required plugins and Javascript dependencies:

\$./install.sh

Windows Users

Similarly, Windows users should execute:

\$ install.bat

Plugins

Build your project for the first time. This will create the [www] folder which is the actual cordova directory and where the plugins will be installed.

\$ grunt build

Follow the same process as with "Libraries" and install the required plug ins by using the command that follows:

\$ cordova plugin add {plugin name or url}

eg

\$ cordova plugin add cordova-plugin-globalization

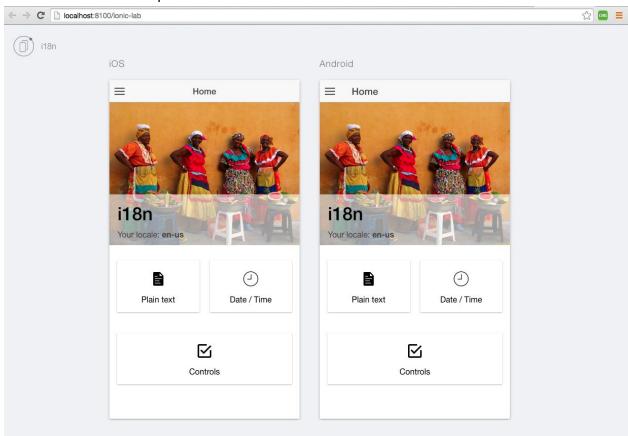
Please check the Readme.md file in the project folder for the list with all the required plugins the application is based on and should be installed.

Run a local development server

Start the local NodeJS server and run the application in the browser:

\$ grunt serve --lab

A browser window will open with two virtual devices the one next to the other.



You could also open the application in a single browser window by starting it with the command:

\$ grunt serve

Run in the emulator

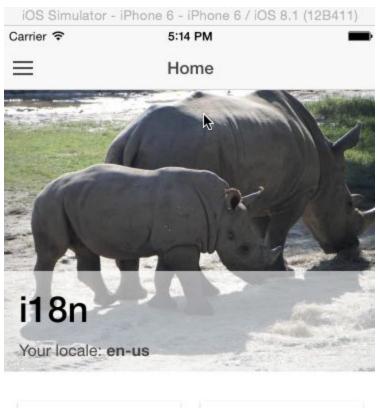
First the prefered platform should be added. In this case iOS:

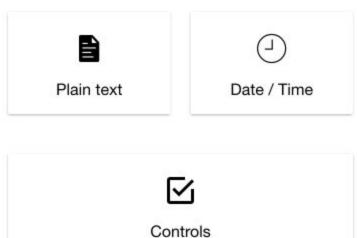
\$ grunt platform:add:ios

Now the application is ready to start inside a simulator:

grunt emulate:ios --livereload

The iPhone simulator will launch and the i18n app will start.





How Internationalisation works

Introduction

In many cases, there is the need to localise your application according to a language or culture

which is called Internationalisation (i18n). This can be done by abstracting all the strings and other locale-specific bits, such as date, out of the application which is called localization (I10n). You can localise your application by including the corresponding locale JS file. For more information, visit https://docs.angularjs.org/guide/i18n.

To set the locale programmatically <u>Angular Dynamic Locale</u> is used. For this, you need to have available the corresponding <code>angular-locale_{flocale}</code>.js (eg <code>angular-locale_en-us.js</code>), for Angular Dynamic Locale to use it without the need to follow the OS locale settings.

As for translation, <u>Angular Translate</u> module is used which utilizes the keys representing the related translation ids.

Configuration

In order to fetch the corresponding locale configuration, you need to add the <u>Cordova</u> <u>Globalization Plugin</u>. Thus, it should be included in the package.json file as shown below:

```
a package.json - /Users/sofia/Documents/i18n - Atom
          package.json
         "node": ">=0.10.0"
50
      },
51
      "scripts": {
        "test": "grunt test",
        "postinstall": "./install.sh"
54
      },
      "cordovaPlatforms": [
        "ios",
        "android"
      ],
      "cordovaPlugins": [
60
         "org.apache.cordova.device",
        "org.apache.cordova.console",
        "com.ionic.keyboard",
         "cordova-plugin-whitelist",
        "cordova-plugin-globalization"
64
   }
package.json 47:5 (9)
                                                                         UTF-8 JSON № master
```

package.json

Also, the dependency for Angular Dynamic Locale needs to be injected by adding the highlighted area in app.js file under the app/scripts path as shown below:

```
app.js - /Users/sofia/Documents/i18n - Atom
  v is scripts
                                      en-US.json
                                                                app.js
                                                                                        date-time.controller.js
                                    che zna parameter Lo un uniuj or reguzito
    > common
                              6 // 'starter.controllers' is found in controllers.js
    > a controls
                                 angular.module('i18n', [
    > adate-time
                             8
                                   'ionic',
                                                                                                           I
    > home
                             9
                                   'config',
    > infrastructure
                             10
                                   'i18n.common',
                                   'i18n.infrastructure',
    > plain-text
                                   'i18n.plain-text',
     app.js
                                   'i18n.home',
     configuration.js
                             14 'i18n.menu',
   styles
                             15 'i18n.controls',
     common.scss
                            16 'i18n.date-time',
     home.scss
                            17 'ngCordova',
     main.scss
                            18 'tmh.dynamicLocale',
   index.html
                                   'pascalprecht.translate'
   I README.md
                            20 ])
> node_modules
                                   .value('_', window._)
> platforms
                                   .constant('availableLanguages', ['en-US', 'ru-RU'])
> plugins
                                   .constant('defaultLanguage', 'en-US')
                             26
                                   .run(function (
> | www
                                     $ionicPlatform, tmhDynamicLocale, $translate, $cordovaGlobalization,
 a.bowerrc
                                     availableLanguages, $rootScope, defaultLanguage, $locale) {
 editorconfig
```

app.js

Additionally, in order to inject Angular Translate you need to add it in the app.js file under the app/scripts path as shown below:

```
app.js - /Users/sofia/Documents/i18n - Atom

▼ ■ scripts

                                      en-US.json
                                                                                         date-time.controller.js
                                                                    app.js
    > m common
                              1 // Ionic Starter App
    > controls
    > adate-time
                              3 // angular.module is a global place for creating, registering and ret
                                 // 'i18n' is the name of this angular module example (also set in a <
    > infrastructure
                              5 // the 2nd parameter is an array of 'requires'
    > menu
                              6 // 'starter.controllers' is found in controllers.js
    > plain-text
                              7 angular.module('i18n', [
     app.js
                              8
                                   'ionic',
                                  'config',
     configuration.js
                              9
                                   'i18n.common',

✓ in styles

     common.scss
                                   'i18n.infrastructure',
                             12 'i18n.plain-text',
     home.scss
                                    'i18n.home',
     main.scss
                                    'i18n.menu',
                             14
   index.html
                                    'i18n.controls',
   README.md
                                    'i18n.date-time',
                                   'ngCordova',
> node_modules
                             18 'tmh.dynamicLocale',
> platforms
                                    'pascalprecht.translate'
> plugins
                             20 ])
> resources
> test
                                    .value('_', window._)
> www
```

app.js

Configuring a new language

Two folders are responsible for the Internationalisation of this application: i18n and locales. The former is used for the translation of plain text and the latter for the localisation of the application in the specific region's date/time settings.

For more details on how to add a new language/region settings, please see section "Add a new language/locale".

Language files and Translations

For the translation files, you need to create a JSON file under the app/il8n path consisting of the keys representing the related translation ids which looks similar to the following:

```
a en-US.json - /Users/sofia/Documents/i18n - Atom
🕶 📮 i18n
                                         app.js
                                                                  en-US.json
                                                                                        angular-locale_el-gr.js
  > ■ .git
                               1 {
  > iii .sass-cache
                                     "CONTROLS": "Controls",
  > in .settings
                                    "USERNAME": "Username",
                                   "PASSWORD": "Password",
  > 🛅 .temp
                                   "COUNTRY": "Country",
  🕶 🛅 арр
                                   "REGISTER": "Register",
                               6
    > bower components
                                   "USA": "USA",
    > fonts
                                   "GREECE": "Greece",
    ♥ i18n
                                   "UKRAINE": "Ukraine",
                               9
       el-GR.json
                              10 "FRANCE": "France",
       en-US.json
                                  "PLAIN_TEXT": "Plain text",
       ru-RU.json
                                  "LOREM_IPSUM": "Lorem ipsum",
    > images
                                   "LOREM_IPSUM_FULL": "Lorem ipsum dolor sit amet, consectetur adipisicing elit,
    > locales
                                  "DATE_TIME": "Date / Time",
                              14

▼ ■ scripts

                                  "HOME": "Home",
      > common
                              16 "YOUR_LOCALE": "Your locale"
      > controls
                              17 }
      > a date-time
      > home
      > infrastructure
      > menu
```

en-US.json

As shown above, each JSON file corresponds to a specific language and contains all the translation keys(red colored) followed by the respective text in a particular language(green colored).

As for the files which are used for the date/time locale, i18n expects the angular locales to be present at /i18n/angular-locale_{ {locale}}.js. This file is available for many different languages and can be found in the <u>Angular's repo</u>. Angular's repo contains all the currently supported locales, so you could download the files you need from there.

The /i18n/angular-locale_{{locale}}.js file should be contained under the app/locales path:

```
angular-locale_en-us.js - /Users/sofia/Documents/i18n - Atom
→ 📙 i18n
                                                                  angular-locale_en-us.js
  agit .git
                                1 'use strict';
  > asss-cache
                                angular.module("ngLocale", [], ["$provide", function($provide) {
  > a.settings
                                3 var PLURAL_CATEGORY = {ZERO: "zero", ONE: "one", TWO: "two", FEW:
  > in .temp
                               4 function getDecimals(n) {
 ♥ app
                                   n = n + 11;
    > bower_components
                                    var i = n.index0f('.');
    > fonts
                               7 return (i == -1) ? 0 : n.length - i - 1;
    > i18n
                               8 }
    > images
                               9

▼ Incales

                               10 function getVF(n, opt_precision) {
       angular-locale_el-gr.js
                               var v = opt_precision;
       angular-locale_en-us.js
                               if (undefined === v) {
       angular-locale_ru-ru.js
                               14
                                     v = Math.min(getDecimals(n), 3);

▼ im scripts

                               15 }
      > common
      > controls
                               var base = Math.pow(10, v);
      > adate-time
                                   var f = ((n * base) | 0) % base;
      > home
```

angular-locale_en-us.js

Language tags / codes

Language tags/codes are used in order to identify the language of a text or the language/culture code of the application.

Naming language codes

Some of the language codes that are used in **i18n lonic** application are "en-US", "el-GR", "ru-RU".

The best practices regarding the codes-subcodes creation are illustrated in <u>W3C's</u> <u>Internalization guidelines</u>. Note that, in W3C's Internalization guidelines, codes are referred to as tags. Following the standard W3C recommends, the language codes are the same for all programming languages and they suggest a common interpretation by all professionals as most locale-aware applications conform to this standard.

Alternative locale creation

In case of using different locales from those existing in the <u>Angular's repo</u>, a unicode converter, such as <u>Brenah's Unicode Converter</u>, can be utilised for the representation of non latin characters of language tags.

A working example

In this section, an example, that demonstrates all the aforementioned principles followed by the **i18n lonic** application, is presented.

Home

home.html uses translation keys instead of string values which are defined in translation files under the app/il8n path. A JSON file example for US-english is already shown above (Language Files and Translations section). The translation keys used on the "Home" screen are found in home.html file under the app/scripts/home path as highlighted below:

```
nome.html - /Users/sofia/Documents/i18n - Atom
🕶 📮 i18n
                                                                               home.controller menu-items.ser
                            app.js
                                        ru-RU.ison
                                                      en-US.ison
                                                                    home.html
  > 🛅 .git
                         1 <ion-view view-title="{{ 'HOME' | translate }}">
                         2 <ion-content>
  > asss-cache
                              <div class="home-hello">
  > in .settings
                         4
                                 <img src="http://lorempixel.com/1024/768/"/>
  > 🛅 .temp
                                 <div class="item item-text-wrap">
  ♥ ■ app
    > bower_components
                                                                                                                 I
                                    <div>{{ 'YOUR_LOCALE' | translate }}: <b>{{vm.lang}}</b></div>
    > fonts
                                  </div>
    > ill i18n
                         Q.
                                </div>
    > images
                        10
    > locales
                               <div ng-repeat="e in vm.entries" ng-if="$index%2==0" class="row">
    scripts
      > iii common
                                <div class="card col home">
      > controls
                                   <a class="item item-text-wrap" href="#/app/{{vm.entries[$index].path}}">
                         14
      > adate-time
                                      <h1><i class="icon {{vm.entries[$index].icon}}"></i></h1>
       home
                                      <h2>{{vm.entries[$index].title | translate}}</h2>
          home.controlle
                                    </a>
         home.html
                                   </div>
          home.module.j
          menu-items.se
                             <div class="card col home" ng-if="!!vm.entries[$index + 1]">
                        20
      > infrastructure
                                     <a class="item item-text-wrap" href="#/app/{{vm.entries[$index + 1].path}}">
      > menu
                                      <h1><i class="icon {{vm.entries[$index + 1].icon}}"></i></h1>
                                      <h2>{{vm.entries[$index + 1].title | translate}}</h2>
      > plain-text
                        24
                                    </a>
        app.js
                                   </div>
        a configuration.js
    > styles
                                 </div>
      index.html
                        28
      README.md
                              </inn-content>
  > hooks
                        30 </ion-view>
  > node_modules
```

home.html

It should be noted the fact that the keys of the buttons which are displayed on the "Home" screen are not set directly as the first highlighted key but they are set in

menu-items.service.js under the app/scripts/home path as shown here:

```
000
                                     menu-items.service.js - /Users/sofia/Documents/i18n - Atom
🕶 📮 i18n
                                                                                   home.html
                                                                                                menu-items.servi
  .git
                                      (function() {
  > in .sass-cache
                                        'use strict';
  > in .settings
                                  4
                                       angular
  > iii .temp
                                          .module('i18n.home')
  🕶 🛅 арр
                                  6
                                          .factory('menuItems', menuItems);
    > m bower_components
    > fonts
                                  8
                                       menuItems.$inject = [];
     > i18n
                                 9
     > images
                                 10
                                       /* @ngInject */
     > locales
                                      function menuItems() {

▼ im scripts

                                         var data = [{
       > a common
                                           title: 'PLAIN_TEXT',
       > controls
                                 14
                                            path: 'plain-text',
       > adate-time
                                           icon: 'ion-document-text'

✓ im home

                                        }, {
           home.controller.js
                                           title: 'DATE_TIME',
          home.html
                                            path: 'date-time',
          home.module.js
                                           icon: 'ion-ios-clock-outline'
          menu-items.service.js
                                 20
        infrastructure
                                           title: 'CONTROLS',
                                            path: 'controls',
       > menu
                                            icon: 'ion-android-checkbox-outline'
       > plain-text
                                 24
        app.js
        configuration.js
                                          return data;
     > styles
                                       }
      index.html
                                 28 })();
      README.md
    hooks
```

menu-items.service.js

Plain Text

Similarly, in case that you need to translate a free text, as is the case presented when clicking the "Plain Text" button, you need to utilise a key for the title and one for the text body in the plain-text.html file under the app/scripts/plain-text path:

```
plain-text.html - /Users/sofia/Documents/i18n - Atom
▼ 🛄 i18n
                                                       plain-text.html
                                                                      ru-RU.json
                                                                                              en-US.json
                                 1 <ion-view view-title="{{ 'PLAIN_TEXT' | translate }}">
 > iii .git
                                      <ion-header-bar class="bar-subheader bar-positive">
 > isass-cache
                                       <h1 class="title">{{ 'LOREM_IPSUM' | translate }}</h1>
 > in .settings
                                 4 </ion-header-bar>
 > iii .temp
                                5 <ion-content class="padding" has-subheader="true">
 → 🛅 арр
                                        {{ 'LOREM_IPSUM_FULL' | translate}}
                                6
    > bower_components
                                     </ion-content>
    > fonts
                                8 </ion-view>
    > i18n
    > images
    > locales

▼ im scripts

      > iii common
      > a controls
      > a date-time
      > home
      > infrastructure

▼ ■ plain-text

         plain-text.html
          plain-text.module.js
       app.js
```

plain-text.html

Date/Time

Date/Time formatting is undertaken by Angular as the

/i18n/angular-locale_{{locale}}.js file dictates. For example, this JS file for US-english is angular-locale_en-us.js under the app/locales path as shown in Language Files and Translations section.

Controls

If translating a form is the case, then this scenario is demonstrated when clicking on the "Controls" button. First, in the controls.html file under the app/scripts/controls path, you need to use a key for every form field, every option of the drop-down list and the submit button as highlighted below:

```
ontrols.html - /Users/sofia/Documents/i18n - Atom
🗸 🛄 i18n
                                                                                             controls.html
                                                     ru-RU.ison
                                                                         en-US.ison
                            1 <ion-view view-title="{{ 'CONTROLS' | translate}}">
 > 🛅 .git
                                <div class="content has-header">
 > ass-cache
                                  <div class="list">
 > in .settings
                                   <label class="item item-input">
 > in .temp
                                    <span class="input-label">{{ 'USERNAME' | translate}}</span>
 ♥ mapp
                                     <input type="text">
                            6
   > bower_components
                                  </label>
<label class="item item-input">
   > fonts
                            8
   > ill il8n
                                    9
   > images
                           10
                                      <input type="password">
   > locales
                                  </label>
<div class="item item-input item-select">

▼ image scripts

     > common
                                     <div class="input-label">{{ 'COUNTRY' | translate}}</div>

▼ ■ controls

                                    <select>
        controls.html
                                      <option>{{ 'USA' | translate}}</option>
        controls.module.js
                                      <option>{{ 'GREECE' | translate }}</option>
     > a date-time
                                      <option>{{ 'UKRAINE' | translate }}</option>
     > in home
                                       <option selected="">{{ 'FRANCE' | translate }}</option>
                           18
     > infrastructure
                           19
                                     </select>
     > menu
                           20
                                    </div>
     > plain-text
                                  </div>
                                  <div class="padding">
      app.js
                                    configuration.js
   > styles
                                  </div>
                               </div>
     index.html
                           26 </ion-view>
     III README.md
```

controls.html

Default language

In order to set the default language of the application when no other languages are available, you should modify the highlighted area in the app.js file under the app/scripts path:

```
app.js - /Users/sofia/Documents/i18n - Atom
🕶 📮 i18n
                                          app.js
                               6 // 'starter.controllers' is found in controllers.js
  > 🛅 .sass-cache
                               7 angular.module('i18n', [
  > in .settings
                                     'ionic',
  > 🗎 .temp
                                   'config',
                               9
  → 🛅 арр
                              10 'i18n.common',
    > bower_components
                              11 'i18n.infrastructure',
                                   'i18n.plain-text',
'i18n.home',
    > in fonts
    > 🗎 i18n
                                   'i18n.menu',
    > images
                              14
                                    'i18n.controls',
    > locales
                                    'i18n.date-time',
    16
      > iii common
                                    'ngCordova',
                              18
                                    'tmh.dynamicLocale',
      > a controls
                                     'pascalprecht.translate'
      > adate-time
                              19
                              20 ])
      > home
      > infrastructure
                                     .value('_', window._)
      > menu
      > plain-text
                                     .constant('availableLanguages', ['en-US', 'ru-RU'])
      app.js
                              25
                                   .constant('defaultLanguage', 'en-US')
       configuration.js
                               26
     styles
                                    .run(function (
      index.html
                                     $ionicPlatform, tmhDynamicLocale, $translate, $cordovaGlobalization,
      README.md
                                       availableLanguages, $rootScope, defaultLanguage, $locale) {
```

app.js

Available languages

All the available languages should be declared in the highlighted area in the app.js file under the app/scripts path:

```
● ● ● ▼ 🗒 i18n
                                               app.js - /Users/sofia/Documents/i18n - Atom
                                         app.js
  > 🛅 .qit
                              1 // Ionic Starter App
  > in .sass-cache
  > a .settings
                              3 // angular.module is a global place for creating, registering and retrieving Angular modules
  > im .temp
                              4 // 'i18n' is the name of this angular module example (also set in a <body> attribute in index
  🕶 🛅 арр
                              5 // the 2nd parameter is an array of 'requires'
   > bower_components
                              6 // 'starter.controllers' is found in controllers.js
    > fonts
                              7 angular.module('i18n', [
    > il8n
                              8
                                   'ionic',
    > images
                                 'config',
                              9
    > in locales
                             10 'i18n.common',
   11 'i18n.infrastructure',
                             12 'i18n.plain-text',
13 'i18n.home',
     > common
     > a controls
                             14 'i18n.menu',
      > ate-time
                             15 'i18n.controls',
      > home
                             16 'i18n.date-time',
      > infrastructure
                             'ngCordova',
'tmh.dynamicLocale',
      > menu
      > plain-text
                             19 'pascalprecht.translate'
     app.js
                             20 ])
       configuration.is
     styles
                             22 .value('_', window._)
       common.scss
       home.scss
                             .constant('availableLanguages', ['en-US', 'ru-RU', 'el-GR'])
       main.scss
                                   .constant('defaultLanguage', 'en-US')
     index.html
     III README.md
                                     run(function (
```

app.js

Add a new language/locale

For demonstration purposes, there are three available JSON files under the app/i18n path; one for english, one for russian and one for greek. Naturally, you could add as many languages as you prefer under the same path.

As for the locales, there are three available JS files under the app/locales path; one for english, one for russian and one for greek. You could add as many locales as you prefer under the same path.

More precisely, in order to add a new language/locale you should follow the following steps:

- Create a JSON file defining the translation keys with the corresponding text in a particular language (see <u>Language Files and Translations</u> section).
- Create a JS file defining the date/time format in a particular language/region (see <u>Language Files and Translations</u> section).
- 3. Add the new language in the available languages declared in the app.js file under the app/scripts path (see <u>Available languages</u> section).

Support

With regard to technical questions, new ideas and suggestions, you may use the dedicated form and choose the product your enquiry refers to:

http://support.appseed.io/customer/portal/questions/new

References / Links:

YouTube channel

Periodically, video demonstrations and tutorials related to this product will be published in my YouTube channel.

• Codecanyon User page

You may contact me by using my user page on Codecanyon.

• <u>Titanium Templates Forum</u>

The Google Group that has been created for this product.

Quick Start Guide

The online version of this document.

Thank you

Thank you again for purchasing my product. If you have any questions that are beyond of the scope of this help file, please feel free to email also via my user page contact form. --- Stavros.