Titanium API Doc Style Guide

Overview

The following style proposals are inspired by a number of sources, most notably the JavaDoc style guide in How to Write Doc Comments for the Javadoc Tool. The *Microsoft Guide to Technical Publications* is also a useful reference for general documentation style questions.

Guidelines

Always use initial capital and end punctuation, even if the doc string is a fragment:

- `An object describing the font properties for the label.` (preferred)
- `an object describing the font properties for the label` (avoid)

Object and module summaries should describe the object or module. The creator method should be listed in the description, not the summary.

- `A switch control with on and off states.` (preferred)
- ``A Switch is created by the method `Ti.UI.createSwitch.` `` (avoid)

Start method summaries with an active verb:

- `Pours the whisky.` (preferred)
- `This method pours the whisky.` (avoid)

Use 3rd person declarative instead of 2nd person imperative:

- `Adds a child to this view.` (preferred)
- `Add a child to this view.` (avoid)

When referring to the present object, use "this" not "the."

- ``Multiplies this matrix by the supplied matrix, `m1`.`` (preferred)
- ``Multiplies the matrix by the supplied matrix, `m1`.`` (avoid)

When methods or properties can take a constant value, identify the set of constant values and where they are defined. If the set of possible values is small, the values can be enumerated.

- ``Specify one of the `TEXT_ALIGNMENT` constants from <Titanium.UI>.`` (preferred)
- ``Specify one of `SERVED_ROCKS` or `SERVED_NEAT` from <Titanium.Beverage>.`` (preferred)
- `Specify one of [pe:SERVED_ROCKS](Titanium.Beverage.SERVED_ROCKS) or [pe:SERVED_NEAT] (Titanium.Beverage.SERVED_NEAT).` (preferred)
- `Specify using a constant from <Titanium.UI>.` (avoid)
- ``Specify one of `SERVED_ROCKS` or `SERVED_NEAT`.`` (avoid, unless the constants are defined in the same object/module)

Use code font for keywords and names. This includes:

- JavaScript keywords
- method names
- property names
- parameter names
- module and object names
- variable names
- constants
- code samples

    When used inline, use `backticks` for objects that should be in code font. (When you add inline links to Titanium types, they are automatically set in code font.) For code samples, indent the entire code sample 4 spaces to format as a Markdown code block.

Defining a constant: Need a consistent style here. We should describe both what the constant does, and (if possible) where it's used. We don't include a special annotation for constants, but we do use a consistent style for them (ALL_CAPS). It is not necessary to include the term "constant" in the summary unless it makes the description more clear.

- summary: `Ease-in to the animation. See <Titanium.UI.Animation> for a discussion of animation curves.` (preferred)
- summary: `Constant value representing an "ease-in" animation curve.` (ok)
- summary: `animation curve constant` (avoid)

Using inline links: Inline links are very useful, but excessive use detracts from readability. Use them where they add information not already supplied by the YAML markup. For example, if the return type of a method is Titanium.Blob, the description text can just use code style – `Blob` – instead of an inline link.

```
summary: Returns the image as a `Blob`.
returns:
    type: Titanium.Blob
```

When referring to a given type multiple times in the same description, only one reference (usually the first) should be an inline link.

Using external links: Make sure that the linked content is useful. When linking to content outside of the API docs, introduce the link so users know why they should follow the link. Avoid anonymous inline links.

Preferred:

```
Titanium binding for an Android activity. For more information on Android
activities, see the
[Activity Reference
Page](http://developer.android.com/reference/android/app/Activity.html) on
the Android Developer site.
```

Preferred:

```
For more information on Android Activity, see:

-
[developer.android.com/reference/android/app/Activity.html](http://develop
er.android.com/reference/android/app/Activity.html)}} (preferred)
```

Avoid:

```
Titanium binding for an an [Android
Activity](http://developer.android.com/reference/android/app/Activity.html
).
```

Formatting

For ease of reviewing pull requests on GitHub, TDoc files should be line-wrapped at 100 characters.

For legibility, vertical whitespace may be added before a YAML key that introduces a YAML array, and before each array element:

```
  properties:

    - name: backgroundColor
      summary: Background color for this view.
      type: String

    - name: enabled
      summary: Indicates whether the view is enabled.
      type: Boolean
```

Vertical whitespace should not be used *inside* a YAML array element:

```
    # Avoid this format
    - name: enabled

      summary: Indicates whether the view is enabled.
      type: Boolean
```

JSON-style arrays. Where JSON-style arrays are used (for example, in "platforms:" or when listing multiple types a property can hold), whitespace between items is desirable for ease of reading:

```
    platforms: [android, iphone, ipad]
```

Links

When linking to guides or videos on docs.appcelerator.com from the API docs, use the full URL for the doc site, with the version replaced with `latest` and `index.html` removed from the path. When generating the JSDuck version of the docs for the doc site, these links are translated to local links. For other formats, the link will work as written. Entering the URL in a browser should redirect you to the latest version of the guide.

```
    [Quick
    Start](http://docs.appcelerator.com/platform/latest/#!/guide/Quick_Start)
```

Do not use the versioned link, which will get outdated, or include `index.html`, which breaks the redirection mechanism.

Avoid:

```
[Quick
Start](http://docs.appcelerator.com/platform/2.1/index.html#!/guide/Quick_
Start)
[Quick
Start](http://docs.appcelerator.com/platform/latest/index.html#!/guide/Qui
ck_Start)
```

Inline HTML

Markdown format allows you to use inline HTML. This is especially useful for adding tables, which Markdown doesn't support itself. When adding inline HTML:

- Always use **lowercase** tag names. So use `<table>`, `<code>`, not `<TABLE>`, `<CODE>`.
- All tags must be closed, XHTML style. So use `<br />`, `<td>A Table Cell</a>`.
- When adding an inline table, much formatting is overridden by JSDuck. Use the `doc-table` CSS class to get some reasonable default formatting (`<table class="doc-table">`).

Deprecating and Removing APIs

When deprecating an API, add the `deprecated` tag. You must specify a `since` version. You should specify a `notes` field directing the user to the replacement API or APIs.

```
deprecated:
    since: "7.0.0"
    notes: Use [SOMETHING_ELSE](Titanium.UI.SOMETHING_ELSE) instead.
```

The `removed` version should **NOT** be specified until the API is actually removed.

When the API is removed, add the `removed` version to the `deprecated` tag. **DO NOT** remove the API from the docs.

```
deprecated:
    since: "7.0.0"
    notes: Use [SOMETHING_ELSE](Titanium.UI.SOMETHING_ELSE) instead.
    removed: "8.0.0"
```

Removed APIs should stay in the docs (marked as removed) for a full deprecation cycle. That is, if APIs deprecated in release A are *removed* in release B, the APIs *removed* in release A can be removed from the doc in release B.

Tips and Tricks

Colon (:) characters won't pass validation in normal YAML text fields, and YAML text fields can't start with a backtick (`) character. To include one of these characters in a summary or description field, use the YAML chunking operator (|) to add free-form markdown text:

```
- description: |
      Specify one of: <Titanium.Beverage.WHISKY_ISLAY>,
<Titanium.Beverage.WHISKY_HIGHLAND>, <Titanium.Beverage.WHISKY_LOWLAND>.

- summary: |
      `true` if the current view is awesome.
```

Optional parameters: specify optional method parameters by specifying "`optional: true`" and adding a default value.

```
parameters:
  - name: cubes
    summary: Number of ice cubes.
    type: Number
    optional: true
    default: 0
```

Platform-specific parameters: TDoc (and JSCA) don't allow for a parameter to be platform-specific. If one platform adds a parameter, other platforms must accept or ignore that parameter. Describe platform limitations in the summary.

```
  - name: glass
    summary: Glass type, one of <Titanium.Beverage.GLASS_ROCKS> or
<Titanium.Beverage.GLASS_COCKTAIL>. Only used on Android.
    optional: true
    default: Titanium.Beverage.GLASS_ROCKS
```