Project Configuration File (config.json)

Alloy uses the `config.json` file, located in the project's `app` directory, to specify global values, conditional environment and platform values, and widget dependencies. The configuration file contains the following objects:

Object	Description
global	Contains key-value pairs present for all environments and platforms.
env:development	Contains key-value pairs present for targets built for development, running in either the simulator or emulator.
env:test	Contains key-value pairs present for targets built for testing on a device.
env:production	Contains key-value pairs present for targets built for production, running after a package installation.
os:android	Contains key-value pairs present for targets built for Android.
os:ios	Contains key-value pairs present for targets built for iOS.
os:windows	Contains key-value pairs present for targets built for Windows Phone. Note: Support for Windows 8.1 and Windows Phone SDKs has been deprecated as of SDK 6.3.0.GA and has be removed in SDK 7.0.0.GA.
dependencies	Contains key-value pairs for widget dependencies, where the key is the widget name and the value is the version number of the widget.
autoStyle	Enables the autostyle feature for the entire Alloy project. See Dynamic Styles: Autostyle for more information.
backbone	Select the Backbone.js library to use for Alloy Model and Collection objects. Set to either `0.9.2` (default for backwards compatibility), `1.1.2`. or `1.3.3`.

When mixed together, 'os' values override 'env' values, which override 'global' values. If you want to specify both a platform and environment-specific configuation, combine the 'os' and 'env' values together into one string with the values space separated. These values are accessible during runtime by prefixing the key with `Alloy.CFG`.

Example of a configuration file:

config.json

```
{
    "global": { "foo": 1 },
    "env:development": { "foo": 2 },
    "env:test": { "foo": 3 },
    "env:production": { "foo": 4 },
    "os:ios env:production": { "foo": 5 },
    "os:ios env:development": { "foo": 6 },
    "os:ios env:test": { "foo": 7 },
    "os:android": { "foo": 8 },
    "dependencies": {
        "com.foo.widget": "1.0"
    }
}
```

In the above example, if the application is ran on the iPhone simulator and prints out 'foo' using `Ti.API.info(Alloy.CFG.foo)`, it will return '6.'