Blend Documentation

Release 0.1

Justin Walgran

Sep 27, 2017

Contents

Merge, analyze, and optimize client-side assets for web applications and static web sites.

CHAPTER 1

Introduction

Blend was inspired by the asset pipeline introduced in Ruby on Rails 3.1 (http://guides.rubyonrails.org/asset_pipeline.html)

You can create a configuration file to control how Blend analyzes, merges, and optimizes your files, but you don't have to. Without a configuration file, running `blend` in your project directory will:

- Search for any javascript or css files containing blend formatted require statements.
- Run JSLint on the javascript files.
- Recursively merge required files.
- Write the merged files to the `output` directory.
- Write the merged files to the `output` directory.
- Use the YUI compressor to create minified versions of the files in the `output` directory.

CHAPTER 2

Installation

From the Python Package Index

```
pip install blend
```

From Source

```
git clone git://github.com/azavea/blend.git
cd blend
python setup.py install
```

CHAPTER 3

Usage

The `blend` command line tool is designed to process a directory full of files or individually specified files. Given the following directory structure:

```
project
  lib
    jquery.min.js
  src
    app.js
```

And the following `app.js`:

```
/* app.js */

//= require jquery
var app = {};
```

Running `blend` in the `project` directory will result in the following directory tree structure:

```
project
  lib
    jquery.min.js
  output
    app.js
    app.min.js
  src
    app.js
```

Where the content of `project/output/app.js` is `project/lib/jquery.min.js` merged into `project/src/app.js` and `project/output/app.min.js` is the minified version of `project/output/app.js`

Command Line Options

Output

`-o OUTPUT, --output=OUTPUT`

Where the file output will be written. The default is a directory at the root of the project directory named `output`

Path

`-p PATH, --path=PATH`

A directory to be searched for required files. Multiple directories can specified by repeating the flag. If you do not specify any directory with the PATH flag then only the working directory will be searched for required files.

Skip Working Directory

`-s, --skipcwd`

Exclude the current working directory from the requirement search paths.

> **Warning:** If you do not specify any paths using the -p,–path option and you also specify the -s,–skipcwd option then Blend will have no directories in which to search for required files and will be unable to merge your code.

Specify A Configuration File

`-c, --config`

Specify a JSON confguration file that describes the analyzers and minifiers to be used.

CHAPTER 4

Adding Requirements To Files

For javascript files, Blend uses the same require syntax as the Rails 3.1 asset compiler.:

```
//= require jquery
```

For CSS files, Blend will recognize comment requires:

```
/*= require control */
```

Blend will also recognize `@import url("...")` statements in CSS files and treat them as require statements:

```
@import url("control.css")
```

Note: `require` comments reference other files using a "base name." You **must not** include the file extension or version numbers or `min` suffixes. Blend strips these suffixes out of the file name when searching for files to satisfy a requirement.

CHAPTER 5

How Blend Satisfies Requirements

By default, Blend recursively searches in the directory from which it is run for files to satisfy requirements. You can suppress this behavior with the `-s, --skipcwd` argument and you can add additional search paths using the `-p, --path` argument.

Searching By Base Name

Given the following file:

```
/* app.js */

//= require jquery
var app = {};
```

When you run:

```
blend app.js
```

Blend will search for a file with a base name of `jquery` and a `.js` extension. **Any** of the following files would satisfy this requirement:

```
JQUERY.js
jquery.javascript
jQuErY.js
jquery-1.2.3.js
JQUERy.min.js
jquery-1.2.3-min.js
```

None of these files would satisfy this requirement:

```
jjquery.js
jquery123.js
```

```
jquery.jscript
jquery-minified.js
```

Search Priority

Blend will prefer to satisfy a requirement with a file in the same directory as the requiring file or a subdirectory of the requiring file. Given this `app.js`:

```
/* app.js */

//= require component
var app = {};
```

and this directory structure:

```
project
  lib
    component.js
  src
    app.js
    components
      component.js  <- this file will be merged with app.js
```

The `component.js` file nested under `src/components` will "win."

CHAPTER 6

Analyzers

By default, Blend runs JSLint on all the javascript files it processes. This can generate failures when you are merging in 3rd party libraries that do not pass JSLint. To get around this problem, the default JSLint analyzer is configured to ignore any files that are under a `lib` directory at the root of the project folder.

Given the following `app.js`:

```
/* app.js */

//= require jquery
var app = {};
```

Running `blend` in this directory structure:

```
project
  src
    app.js
    jquery.min.js
```

Will fail because the minified JQuery library will not pass a JSLint check. However, if you move the JQuery file so the directory structure looks like this:

```
project
  lib
    jquery.min.js
  src
    app.js
```

Then running `blend` will succeed because the JSLint analyzer will skip over `project/lib/jquery.min.js`.

CHAPTER 7

Configuring Blend

Blend can read configuration options from a JSON formatted file. Here is what the default configuration looks like:

```
{
  "analyzers": {
    "javascript": [
      {
        "name": "blend.JSLintAnalyzer",
        "skip_list": [
          "bin"
        ]
      }
    ]
  },
  "minifiers" {
    "javascript": {
      "name": "blend.YUICompressorMinifier"
    },
    "css": {
      "name": blend.YUICompressorMinifier"
    }
  }
}
```

Blend can load configuration files in two ways:

1. From `{current working directory}/.blend/config.json`
2. From a file specifed with the `-c,--config` command line argument

A config file specified with the `-c,--config` command line argument will override a `.blend/config.json` file.