

# Dokumentation GAL Web Template

## Inhaltsverzeichnis

Vorraussetzungen.....	2
Installation.....	2
SCSS Kompilierung.....	3
Libraries.....	3
Projekt-Dateistruktur (MVC-like).....	4
CSS Guidelines (How to name CSS Classes).....	7

# Vorraussetzungen

## 1. Ruby

1. Windows <http://rubyinstaller.org/>
2. Sonstige: <https://www.ruby-lang.org/de/downloads/>

Während der Installation die Haken bei  
"Add Ruby executables to your PATH" und  
"Associate .rb and .rbw files with this Ruby installation" setzen.

## 2. Bundler ( Managing Ruby-Gem Dependencies )

1. <http://bundler.io/>
2. oder in Konsole: `gem install bundler`

# Installation

1. Projekt klonen. ( `git.gal:gal-template-web` )
2. Konsole in Projektverzeichnis öffnen und folgenden Befehl eingeben:

```
bundle install
```

Hierdurch werden alle im Gemfile angegebenen Abhängigkeiten geprüft und falls nötig nachinstalliert. (Darunter [Sass](#) und [Compass](#))

# SCSS Kompilierung

Einfachste Methode:

Im Projekthauptverzeichnis per Konsole 'bcw.bat' ausführen oder deren Inhalt selbst eingeben und ausführen:

```
cd <Projektunterverzeichnis>
```

```
bundle exec compass watch
```

Projektunterverzeichnis ist im Template der Ordner 'src', Verzeichniswechsel muss je nach Betriebssystem/angepasster Projektstruktur natürlich angepasst werden.

Alternative Methode:

Programme wie [Koala](#) oder [Prepos](#) zum Kompilieren einrichten.

Das Projekt enthält 2 Compass Konfigurationsdateien, einmal '*config.rb*' und '*config\_prod.rb*'.

Nach erfolgreicher Kompilierung über *config.rb* wird automatisch über *config\_prod.rb* kompiliert um die Development CSS-Datei und die Produktion CSS-Datei zu erstellen. Dies lässt sich bei Bedarf innerhalb der *config.rb* auskommentieren/anpassen.

Produktion CSS = compressed Output, daher Entfernung sämtlicher Kommentare.

## Libraries

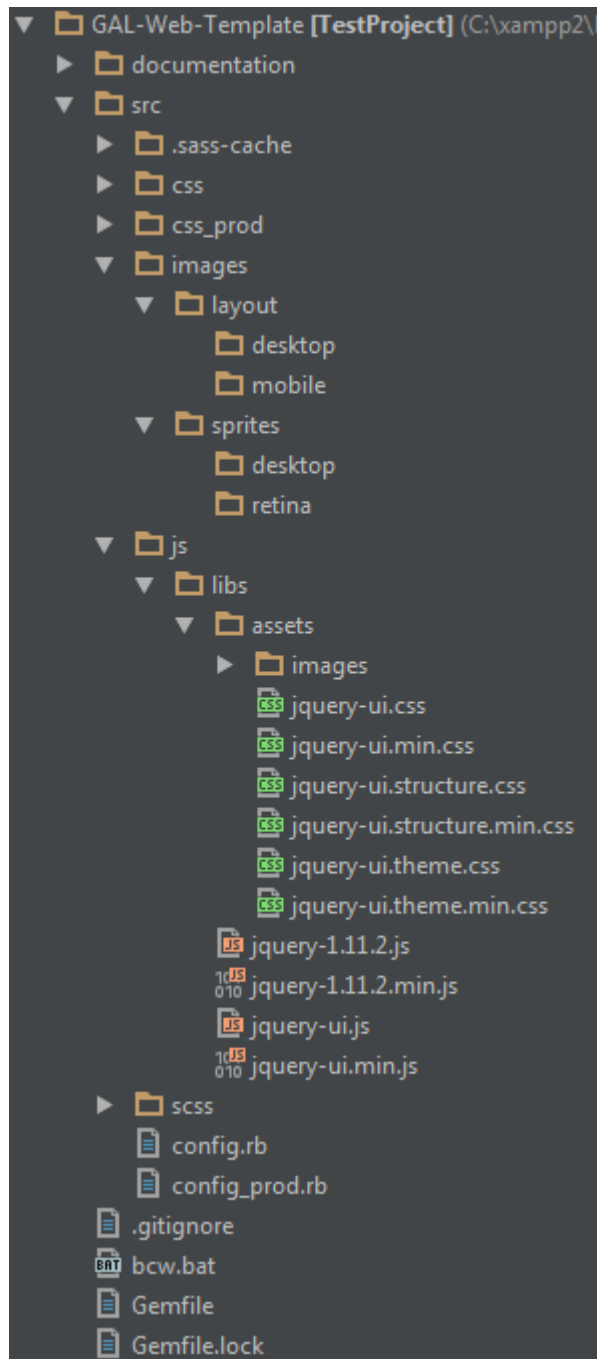
Aktuell im Projekt eingebundene Libraries/Ruby Gems:

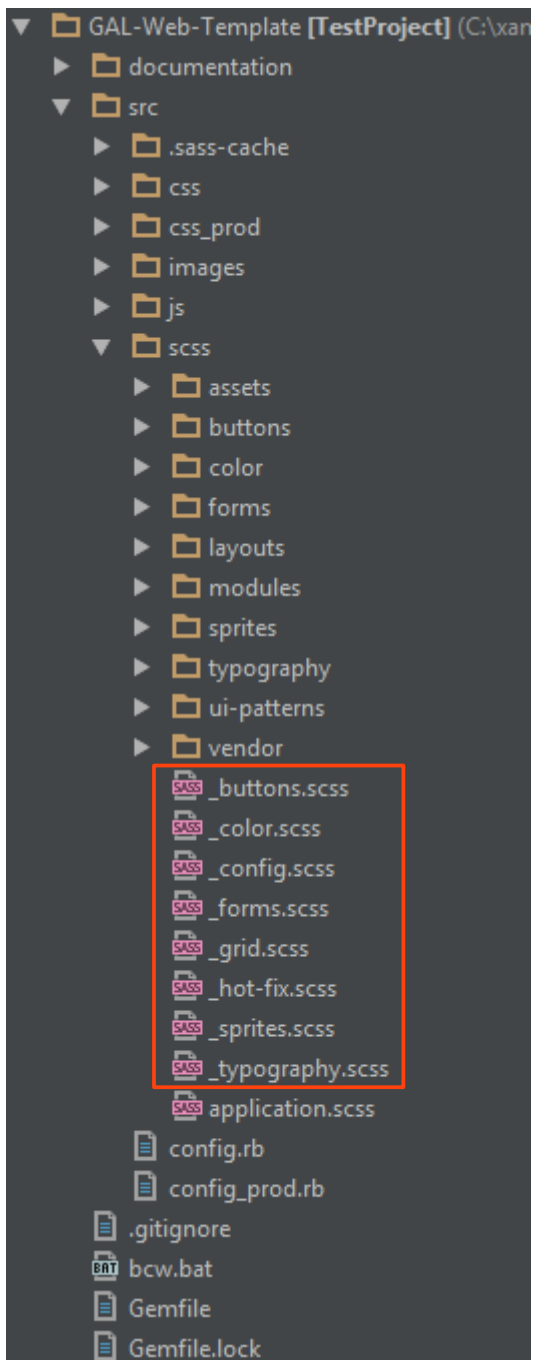
- [Compass](#) ( *CSS/Sass Authoring Framework* )
- [stipe](#) (Compass Extension, Stipe is the core set of Sass Mixins to support the [Toadstool](#) Styleguide Framework )
- [sass\\_globbering](#) (Ermöglicht den Import mehrerer SCSS-Dateien in einem Import-Statement. )
- [normalize-scss](#) (Sass/Compass Version der Normalize.css )
- [sass-css-importer](#) (Ermöglicht den Import von CSS-Dateien in Sass)

Diese Libraries werden Projekt-Extern installiert, sind also global als Ruby-Gems verfügbar.

# Projekt-Dateistruktur (MVC-like)

Die Dateistruktur wurde nach Vorbild von stipe, diesem [Vortrag](#) und ein paar Ideen aus anderen Quellen entworfen.

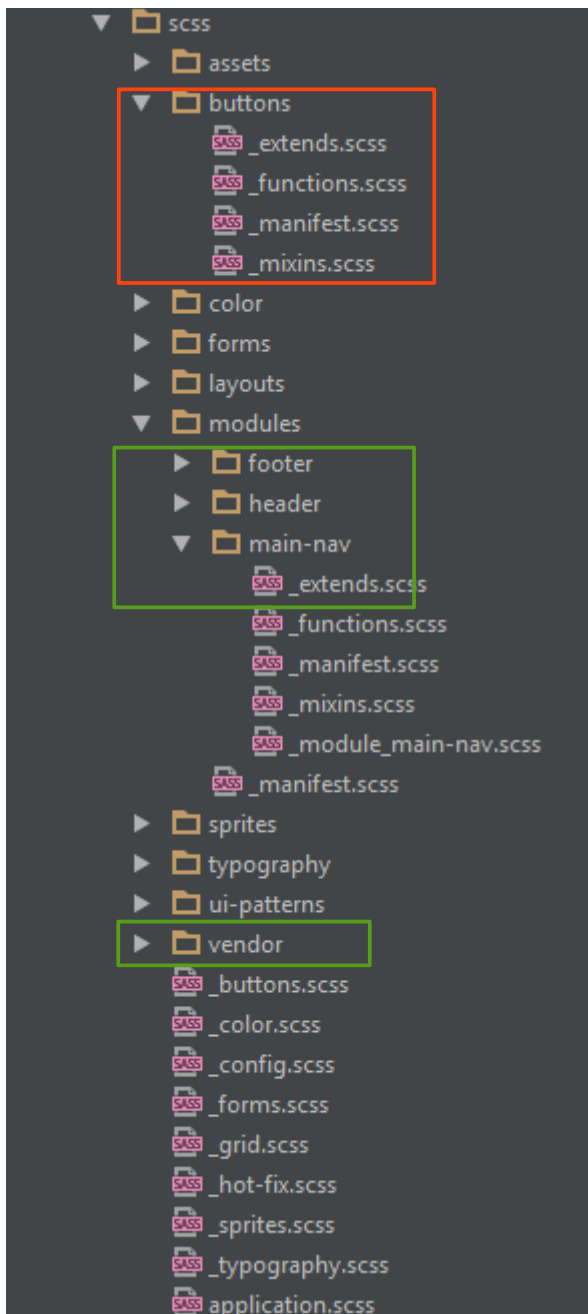




## Übersicht des SCSS Ordners

In diesen Partialen werden Styles untergebracht, keine Logik (keine Pseudoklassen, Mixins oder ähnliches).

Partials = Sass / Scss Dateien mit führendem underscore, hieraus wird keine CSS-Datei generiert.



Partials und zugehörige Ordner sind identisch benannt. In einem solchen Ordner wird die Logik des Partials untergebracht. Pseudoklassen (extends), Funktionen und Mixins.

Modules und UI-Patterns

# CSS Guidelines (How to name CSS Classes)

BEM – meaning *block, element, modifier* – is a front-end naming methodology thought up by the guys at **Yandex**. It is a smart way of naming your CSS classes to give them more transparency and meaning to other developers. They are far more strict and informative, which makes the BEM naming convention ideal for teams of developers on larger projects that might last a while.

[In diesem Artikel näher beschrieben.](#)