2014

KARLSRUHER ENTWICKLERTAG

Grunt, Gulp & fabs

Build-System and Development-Workflow for modern Web-Applications

ABOUT ME

Philipp Burgmer

- Software Engineer / Consultant / Trainer
- Focus: Frontend, Web Technologies

- WeigleWilczek GmbH
- burgmer@w11k.com

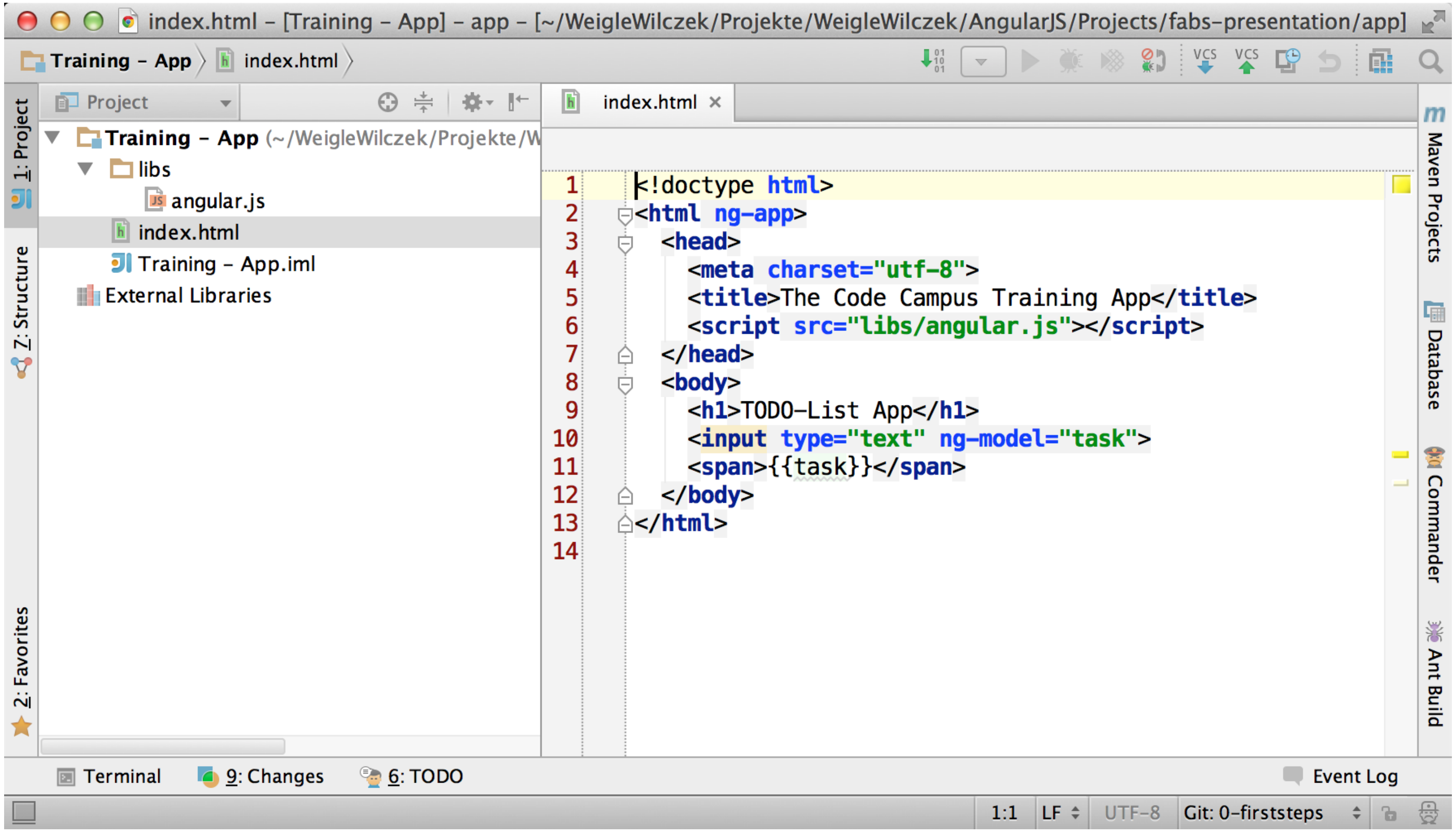
ABOUT US

WeigleWilczek / W11k

- Software Design, Development & Maintenance
- Consulting, Trainings & Project Kickoff

- Web Applications with AngularJS
- Native Rich Clients with Eclipse RCP

HOW DO YOU START A NEW WEB-APP?

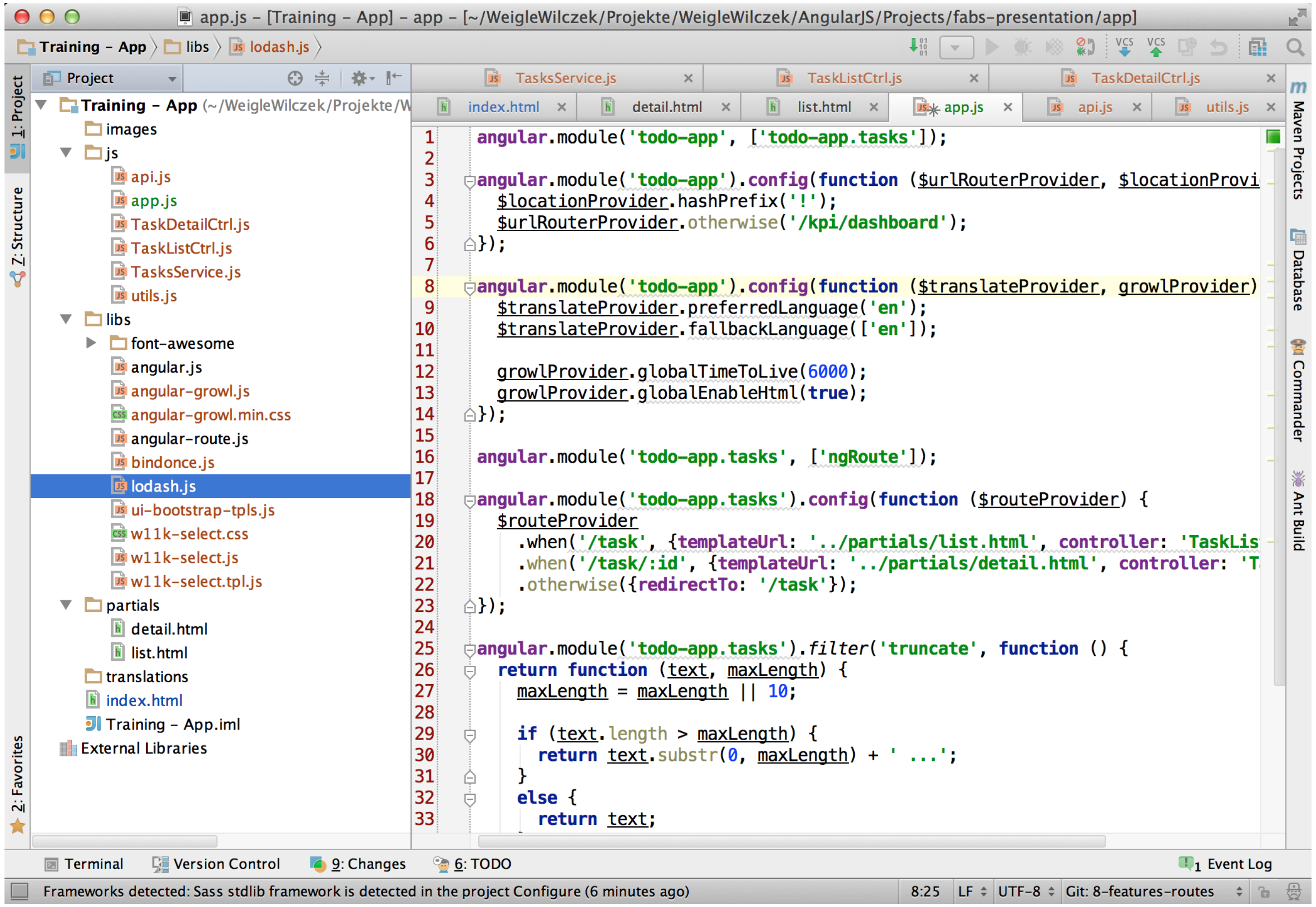


```html
<!doctype html>
<html ng-app>
  <head>
    <meta charset="utf-8">
    <title>The Code Campus Training App</title>
    <script src="libs/angular.js"></script>
  </head>
  <body>
    <h1>TODO-List App</h1>
    <input type="text" ng-model="task">
    <span>{{task}}</span>
  </body>
</html>
```

AND HOW DOES IT LOOK LIKE AFTER ONE NIGHT OF

edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, edit, switch, refresh, test, switch, ...

A LOT OF FILES TO MANAGE



app.js – [Training – App] – app – [~/WeigleWilczek/Projekte/WeigleWilczek/AngularJS/Projects/fabs-presentation/app]

```
Training – App (~/WeigleWilczek/Projekte/W
    images
    js
        api.js
        app.js
        TaskDetailCtrl.js
        TaskListCtrl.js
        TasksService.js
        utils.js
    libs
        font-awesome
        angular.js
        angular-growl.js
        angular-growl.min.css
        angular-route.js
        bindonce.js
        lodash.js
        ui-bootstrap-tpls.js
        w11k-select.css
        w11k-select.js
        w11k-select.tpl.js
    partials
        detail.html
        list.html
    translations
    index.html
    Training – App.iml
External Libraries
```

```
angular.module('todo-app', ['todo-app.tasks']);

angular.module('todo-app').config(function ($urlRouterProvider, $locationProvi
  $locationProvider.hashPrefix('!');
  $urlRouterProvider.otherwise('/kpi/dashboard');
});

angular.module('todo-app').config(function ($translateProvider, growlProvider)
  $translateProvider.preferredLanguage('en');
  $translateProvider.fallbackLanguage(['en']);

  growlProvider.globalTimeToLive(6000);
  growlProvider.globalEnableHtml(true);
});

angular.module('todo-app.tasks', ['ngRoute']);

angular.module('todo-app.tasks').config(function ($routeProvider) {
  $routeProvider
    .when('/task', {templateUrl: '../partials/list.html', controller: 'TaskLis
    .when('/task/:id', {templateUrl: '../partials/detail.html', controller: 'T
    .otherwise({redirectTo: '/task'});
});

angular.module('todo-app.tasks').filter('truncate', function () {
  return function (text, maxLength) {
    maxLength = maxLength || 10;

    if (text.length > maxLength) {
      return text.substr(0, maxLength) + ' ...';
    }
    else {
      return text;
```

Frameworks detected: Sass stdlib framework is detected in the project Configure (6 minutes ago) | 8:25 | LF | UTF-8 | Git: 8-features-routes

and no structure in sight

AFTER 3 MONTHS

- 43 Dependencies to other Libraries
- 169 JavaScript Files
- 112 HTML Templates
- ...

PROBLEMS

- Manage Dependencies to 3th Party Libraries (Versioning, Conflicts, Download and Inclusion)
- Include Own JavaScript and CSS Files
- Structure and Modularize Code
- Too Much Overhead During Development
- Find Even Simple Errors Only at Runtime

SOLUTION?

THE RIGHT TOOLS MAKE ALL THE DIFFERENCE!

(http://www.nodejs.org)

Runtime for
Dev Tools

(http://www.gruntjs.com)

Task Runner

Bower

(http://www.bower.io)

Dependency
Management

BOWER

A PACKAGE MANAGER FOR THE WEB

- Command Line Tool
- Node Package
- Install Binary via `npm install -g bower`
- No Project Specific Version

BOWER

A PACKAGE MANAGER FOR THE WEB

- Define Dependencies in `bower.json`

```
{
  "name": "fabs-presentation",
  "dependencies": {
    "angular": "1.2.16",
    "twbs-bootstrap-sass": "3.1.1",
    "w11k-slides": "0.4.1"
  }
}
```

- Download Dependencies via `bower install`
- Half the Problem Remains: bower Does Not Manage Inclusion
- Hint: Use Fixed Versions Within a Project and Variable Within a Library

GRUNT

THE JAVASCRIPT TASK RUNNER

- Command Line Tool
- Lets You Automate Almost Everything
- Split into 2 Node Packages
- Install Binary Globally via `npm install -g grunt-cli`
- Define Dependency to Project Specific Version in `package.json`
- Install Locally via `npm install`

GRUNT

THE JAVASCRIPT TASK RUNNER

- Configure Your Tasks and Targets in `Gruntfile.js`
- Everything Is a Plugin (via NPM and `package.json`)
- Countless Plugins Available
- Tasks and Configuration Are Pure JavaScript
- Register Custom Tasks in `Gruntfile.js`,
  No Need to Write a Plugin

SHORTEN DEV-LIFECYCLE

WITH LIVERELOAD

```
var grunt = require('grunt');

module.exports = function () {
  grunt.loadNpmTasks('grunt-contrib-watch');
  grunt.loadNpmTasks('grunt-contrib-connect');

  grunt.registerTask('dev', ['watch', 'connect:dev']);
  grunt.registerTask('default', ['dev']);

  grunt.initConfig({
    watch: { src: { files: ['js/**/*.js', '!js/**/*.test.js'] }},
    connect: { dev: { options: { port: 9000 }}}});
};
```

INCLUDE JAVASCRIPT FILES

WITH CUSTOM TASK

```
var indexHtmlTask = function () {
  var jsFiles = utils.filterForJS(this.filesSrc);

  grunt.file.copy('src/index.html', 'build-output/index.html', {
    process: function (contents) {
      return grunt.template.process(contents, { data: { scripts: jsFiles }});
    }
  });
};

grunt.registerMultiTask('indexHtml', 'Process index.html template', indexHtmlTask);
```

```
<% scripts.forEach(function ( file ) { %>
<script type="text/javascript" src="<%= file %>"></script>
<% }); %>
```

```
indexHtml: {
  dev: { files: ['js/**.*.js', '!js/**/*.test.js'] }
}
```

JSHint

A JAVASCRIPT CODE QUALITY TOOL

- Substitute for Compiler
- Syntax Check
- Coding Rules Validation
- Static Code Analysis

```
watch: {
  src: {
    files: ['js/**/*.js', '!js/**/*.test.js'],
    tasks: ['jshint:src']
  }
}
```

BUT WHAT ABOUT

- Testing
- SASS and/or LESS Support
- Using Mock-Data During Development
- Running Frontend with Real Backend

NO PROBLEM

THERE ARE PLUGINS FOR EVERYTHING

- Karma & grunt-karma for Unit/Spec Tests
- Protractor & grunt-protractor for End-2-End Tests
- grunt-contrib-sass & grunt-contrib-less

- NPM Packages starting with grunt-*
- Plugin Overview at gruntjs.com/plugins (http://gruntjs.com/plugins)
- More than 2800 Plugins, but a Lot of Duplicates

THERE IS STILL MORE TO DO

BEFORE WE CAN DEPLOY OUR APP

- Concatenation
- Minification
- Cache Busting
- Running Tests Against Minified Code

THE REAL PROBLEMS

- Project Specific Build
- Huge `Gruntfile.js`
- Task Dependencies
- Hard to Do It Right
- Same Problems as with Ant

FABS

FABULOUS ANGULARJS BUILD SYSTEM

- github.com/w11k/fabs (https://github.com/w11k/fabs)
- Inspired by ng-boilerplate (https://github.com/ngbp/ngbp)
- Based on Our Best Practices
- Configurable
- Extendable via Lifecycle-Hooks
- Grunt Based

FEATURES

- Manage Frontend Dependencies with Bower
- Dev-Mode with Server, Proxies and LiveReload
- SASS and LESS Support
- Spec and End-2-End Tests
- Mocks for Tests and Dev-Mode
- Project- and Developer-Configuration
- Build and Serve Distribution incl. Cache Busting

DEMO

- fabs-boilerplate (https://github.com/w11k/fabs-boilerplate)
- This Presentation
- Real World Project
- fabs (https://github.com/w11k/fabs)

FABS & GRUNT PROBLEMS

- Highly Interdependent Tasks
- Hard to Maintain
- Hard to Extend (Even With Hooks)
- Grunt Is Slow (Slower than ...)

GULP.JS

- gulpjs.com (http://gulpjs.com)
- Uses Streams Instead fo Temp Files
- Runs Independent Tasks 'Parallel'
- Code Over Configuration
- Node Package: Install with `npm install -g gulp`
- Plugins Are Regular Node Packages

PROCESS JAVASCRIPT WITH GULP

```
var gulp = require('gulp'),
    rename = require('gulp-rename'),
    concat = require('gulp-concat'),
    uglify = require('gulp-uglify');

gulp.task('process-scripts', function() {
  return gulp.src('src/scripts/*.js')
    .pipe(concat('1.0.1/main.js'))
    .pipe(gulp.dest('dest/scripts/'))
    .pipe(rename({suffix: '.min'}))
    .pipe(uglify())
    .pipe(gulp.dest('dest/scripts/'))
});

gulp.task('watch', function() {
  gulp.watch('src/scripts/*.js', ['process-scripts'])
});
```

CONCLUSION

- A Lot of Small Tools / Plugins, Each for a Specific Task
- Difficult to Keep Track of All These Tools / Plugins
- No De-Facto Standard Build Tool like Maven for Java
- Impossible to Write One for All Web-Apps
- fabs for AngularJS Applications
- Working on a Gulp Port of fabs

2014

KARLSRUHER
ENTWICKLERTAG

MEET THE SPEAKER

@ Speaker Lounge
Second Floor Directly About Reception

Philipp Burgmer
burgmer@w11k.com

www.w11k.com (http://www.w11k.com)
www.thecodecampus.de (http://www.thecodecampus.de)