High performance. Delivered.

Scaffolding in JavaScript

March 2015

Tomi Vanek

accenture

Strategy | Digital | Technology | Operations

Tomi Vanek

Senior technology architect by Accenture

25+ years of experience

Current focus on modern web applications

https://www.youtube.com/watch?v=2nMD6sjAe8I#t=134

Credit: Flickr, photo by Cedward Brice

Initialization

Interaction

Configuration

Generation

Installation

End

- name must be prefixed by generator-.
- folder tree reflects available generators

```
generator-angular
|
|__ package.json
|
|__ generators
|     |__ app
|     |     |__ index.js
|     |
|     |__ controller
|     |     |__ index.js
|     |
|     |__ directive
|           |__ index.js
|
|__ templates
```

default generator

Initialization

Interaction

Configuration

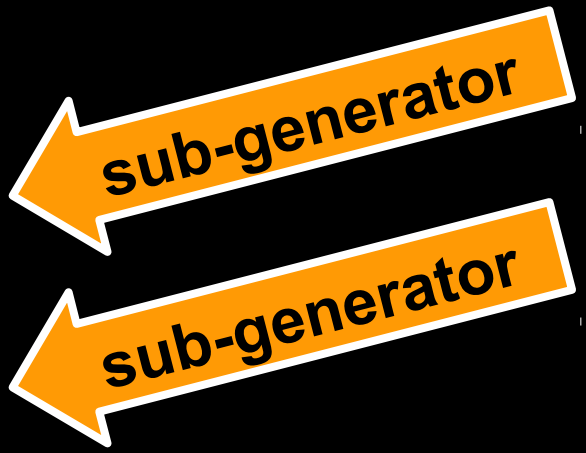
Generation

Installation

End

- name must be prefixed by generator-.
- folder tree reflects available generators

```
generator-angular
|
|__ package.json
|
|__ generators
|     |__ app
|     |     |__ index.js
|     |
|     |__ controller
|     |     |__ index.js
|     |
|     |__ directive
|           |__ index.js
|
|__ templates
```



Initialization

Interaction

Configuration

Generation

Installation

End

- Start with extending from
  - generators.**Base** for default generator
  - generators.**NamedBase** for sub-generator
- Methods/functions are executed in the order they are defined
- Private methods – use underscore or instance method
- Execution phases:
  - **Initializing**
  - **Prompting**
  - **Configuring**
  - **Default**
  - **Writing**
  - **Conflicts**
  - **Install**
  - **End**

Initialization

Interaction

Configuration

Generation

Installation

End

- Start with extending from
  - generators.**Base** for default generator
  - generators.**NamedBase** for sub-generator
- Methods/functions are **executed in the order** they are defined
- Private methods – use underscore or instance method
- Execution phases:
  - **Initializing**
  - **Prompting**
  - **Configuring**
  - **Default**
  - **Writing**
  - **Conflicts**
  - **Install**
  - **End**

Initialization

Interaction

Configuration

Generation

Installation

End

- Start with extending from
  - generators.**Base** for default generator
  - generators.**NamedBase** for sub-generator
- Methods/functions are **executed in the order** they are defined
- Private methods – use **underscore** or instance method
- Execution phases:
  - **Initializing**
  - **Prompting**
  - **Configuring**
  - **Default**
  - **Writing**
  - **Conflicts**
  - **Install**
  - **End**

Initialization

Interaction

Configuration

Generation

Installation

End

- Start with extending from
  - generators.**Base** for default generator
  - generators.**NamedBase** for sub-generator
- Methods/functions are **executed in the order** they are defined
- Private methods – use **underscore** or instance method
- Execution phases:
  - **Initializing**
  - **Prompting**
  - **Configuring**
  - **Default**
  - **Writing**
  - **Conflicts**
  - **Install**
  - **End**

Initialization

Interaction

Configuration

Generation

Installation

End

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({

  constructor: function(args, options) {
    yg.generators.Base.apply(this, arguments);
  },

  greetDeveloper: function() {
    this.log('Hello!');
  },

  createFiles: function() {

  },

  end: function() {

  },

  _helper: function() {

  }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({

  constructor: function(args, options) {
    yg.generators.Base.apply(this, arguments);
  },

  greetDeveloper: function() {
    this.log('Hello!');
  },

  createFiles: function() {

  },

  end: function() {

  },

  _helper: function() {

  }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({

  constructor: function(args, options) {
    yg.generators.Base.apply(this, arguments);
  },

  greetDeveloper: function() {
    this.log('Hello!');
  },

  createFiles: function() {

  },

  end: function() {

  },

  _helper: function() {

  }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({

  constructor: function(args, options) {
    yg.generators.Base.apply(this, arguments);
  },

  greetDeveloper: function() {
    this.log('Hello!');
  },

  createFiles: function() {

  },

  end: function() {

  },

  _helper: function() {

  }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({

  constructor: function(args, options) {
    yg.generators.Base.apply(this, arguments);
  },

  greetDeveloper: function() {
    this.log('Hello!');
  },

  createFiles: function() {

  },

  end: function() {

  },

  _helper: function() {

  }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Arguments
  - yo angular:controller admin

```
var yg = require('yeoman-generator');

module.exports = yg.generators.NamedBase.extend({

 constructor: function(args, options) {
  yg.generators.NamedBase.apply(this, arguments);
 },

 displayName: function() {
     this.log('Creating ' + this.name + 'Ctrl.');
 }

});
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Arguments
  - yo angular:controller admin

```
var yg = require('yeoman-generator');

module.exports = yg.generators.NamedBase.extend({

 constructor: function(args, options) {
  yg.generators.NamedBase.apply(this, arguments);
 },

 displayName: function() {
     this.log('Creating ' + this.name + 'Ctrl.');
 }

});
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Options
  - yo angular --coffee --skip-install

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({

 constructor: function(args, options) {
  yg.generators.Base.apply(this, arguments);
 },
 createFiles: function() {
  if (this.options['coffee']){
    // use CoffeeScript templates
  } else {
    // use JavaScript templates
  }
 },
 install: function() {
  if (this.options['skip-install']) {
   // don't run installation
  }
 }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Options
  - yo angular --coffee --skip-install

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({

 constructor: function(args, options) {
  yg.generators.Base.apply(this, arguments);
 },
 createFiles: function() {
  if (this.options['coffee']){
    // use CoffeeScript templates
  } else {
    // use JavaScript templates
  }
 },
 install: function() {
  if (this.options['skip-install']) {
   // don't run installation
  }
 }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Options
  - yo angular --coffee --skip-install

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({

 constructor: function(args, options) {
  yg.generators.Base.apply(this, arguments);
 },
 createFiles: function() {
  if (this.options['coffee']){
    // use CoffeeScript templates
  } else {
    // use JavaScript templates
  }
 },
 install: function() {
  if (this.options['skip-install']) {
   // don't run installation
  }
 }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
// list
 ? What would you like to write scripts with?:
   > JavaScript
     CoffeeScript

// checkbox
 ? Which modules would you like to include?:
   > (•) angular-animate.js
     ( ) angular-cookies.js
     (•) angular-route.js
     ( ) angular-touch.js

// confirm
? Would you like to include Bootstrap?: (Y/n)

// expand
? Overwrite bower.json?: (Ynaxdh) a
>> overwrite this and all others
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Prompts – via Inquirer.js
  - **Asking questions**
  - **Parsing**
  - **Validation**
  - **Hierarchical prompts**
  - **Error handling**
- Available prompt types
  - **List**
  - **Raw list**
  - **Checkbox**
  - **Confirm (y/n)**
  - **Expand**
  - **Input**
  - **Password**

Initialization

Interaction

Configuration

Generation

Installation

End

```
askAboutAngularModules: function() {
  var ngModules = [{
        value: 'ngAnimate',
        checked: true
      },{
        value: 'ngResource',
        checked: false
      },{
        value: 'ngCookies',
        checked: false }];
  var cb = this.async();
  this.prompt([
   {
       type: 'checkbox',
       name: 'angularModules',
       message: 'Which modules would you like to
include?',
       choices: ngModules
     }
  ], function(answers) {
      this.angularModules = nswers.angularModules;
      cb()
  }.bind(this));
}
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
askAboutAngularModules: function() {
  var ngModules = [{
       value: 'ngAnimate',
       checked: true
     },{
       value: 'ngResource',
       checked: false
     },{
       value: 'ngCookies',
       checked: false }];
  var cb = this.async();
  this.prompt([
   {
      type: 'checkbox',
      name: 'angularModules',
      message: 'Which modules would you like to
include?',
      choices: ngModules
    }
  ], function(result) {
     this.angularModules = result.angularModules;
     cb();
  }.bind(this));
}
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
askAboutAngularModules: function() {
  var ngModules = [{
       value: 'ngAnimate',
       checked: true
     },{
       value: 'ngResource',
       checked: false
     },{
       value: 'ngCookies',
       checked: false }];
  var cb = this.async();
  this.prompt([
   {
      type: 'checkbox',
      name: 'angularModules',
      message: 'Which modules would you like to
include?',
      choices: ngModules
    }
  ], function(answers) {
     this.angularModules = nswers.angularModules;
     cb()
  }.bind(this));
}
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Storing user configuration and sharing it between sub-generators:
  - *Source directory*
  - *Bootstrap*
  - *Routing option*
  - *SASS/LESS*
  - *HTML/Jade*
  - *Path to custom templates*
  - *etc.*

- Configuration Persistence - API
  - **generator.config.save()**
  - **generator.config.set()**
  - **generator.config.get()**
  - **generator.config.getAll()**
  - **generator.config.delete()**
  - **generator.config.defaults()**

Initialization

Interaction

Configuration

Generation

Installation

End

.yo-rc.json – yeoman configuration file

```
{
  "generator-angular": {
        "srcDir": "src/client",
        "routing": "uiRouter",
        "bootstrap": true,
        "ngModules": [
           "ngAnimate",
           "ngSanitize",
           "ngTouch"
        ],
        "extensions": ["js", "html"]
  },
  "generator-node": {
       "srcDir": "src/server",
       "socketio": false,
       "oauth": true
  }
}
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Two location contexts:
  - sourceRoot
  - destinationRoot
- File path is relative to location contexts
- File utilities:
  - template(source, dest [, data])
  - copy(source, dest)
  - dir(source, dest [, data])
- Template data – optional JS object
  - If not provided – properties of `this` are used
- File conflict handling out of box

```
? Overwrite bower.json?: (Ynaxdh) a
  >> overwrite this and all others
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
(function() {
  'use strict';
  angular
   .module('<%= moduleName %>')
   .directive('<%= name %>', [<%= name %>]);

  /* @ngInject */
  function <%= name %> () {
    var directive = {
      restrict: 'EA',
      link: link,<% if (separateTemplate) { %>
      templateUrl: '<%= templateUrl %>'
                 <% } else { %>
      template: '<div><%= name %></div>'<% } %>
    };
    return directive;

    function link(scope, element, attrs) {
      element.text('<%= name %>');
    }
  }
})();
```

Variables

Initialization

Interaction

Configuration

Generation

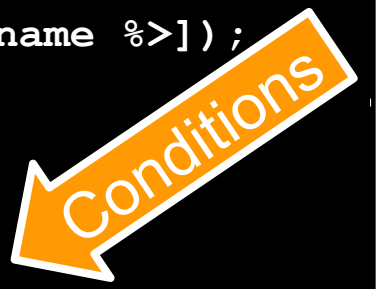
Installation

End

```
(function() {
  'use strict';
  angular
   .module('<%= moduleName %>')
   .directive('<%= name %>', [<%= name %>]);

  /* @ngInject */
  function <%= name %> () {
    var directive = {
      restrict: 'EA',
      link: link,<% if (separateTemplate) { %>
      templateUrl: '<%= templateUrl %>'
                 <% } else { %>
      template: '<div><%= name %></div>'<% } %>
    };
    return directive;

    function link(scope, element, attrs) {
      element.text('<%= name %>');
    }
  }
})();
```



Initialization

Interaction

Configuration

Generation

Installation

End

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({
  constructor: function(args, options) {
    yg.generators.Base.apply(this, arguments);
    this.sourceRoot('../templates');
    this.destinationRoot('.');
    this.appName = 'myApp';
  },

  configFiles: function() {
    this.copy('_editorconfig', '.editorconfig');
    this.dir('code-analysis', '.');
    this.dir('pkg-conf', '.', {name: 'demo'});
  },

  appModule: function() {
    this.template('module/module.js_',
      'src/client/app/app.module.js');
    this.template('controller/ctrl.js_',
      'src/client/app.controller.js',
      {name: 'MyAppCtrl', appName: 'myApp'});
  }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({
  constructor: function(args, options) {
    yg.generators.Base.apply(this, arguments);
    this.sourceRoot('../templates');
    this.destinationRoot('.');
    this.appName = 'myApp';
  },

  configFiles: function() {
    this.copy('_editorconfig', '.editorconfig');
    this.dir('code-analysis', '.');
    this.dir('pkg-conf', '.', {name: 'demo'});
  },

  appModule: function() {
    this.template('module/module.js_',
      'src/client/app/app.module.js');
    this.template('controller/ctrl.js_',
      'src/client/app.controller.js',
      {name: 'MyAppCtrl', appName: 'myApp'});
  }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({
  constructor: function(args, options) {
    yg.generators.Base.apply(this, arguments);
    this.sourceRoot('../templates');
    this.destinationRoot('.');
    this.appName = 'myApp';
  },

  configFiles: function() {
    this.copy('_editorconfig', '.editorconfig');
    this.dir('code-analysis', '.');
    this.dir('pkg-conf', '.', {name: 'demo'});
  },

  appModule: function() {
    this.template('module/module.js_',
      'src/client/app/app.module.js');
    this.template('controller/ctrl.js_',
      'src/client/app.controller.js',
      {name: 'MyAppCtrl', appName: 'myApp'});
  }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

```
var yg = require('yeoman-generator');

module.exports = yg.generators.Base.extend({
  constructor: function(args, options) {
    yg.generators.Base.apply(this, arguments);
    this.sourceRoot('../templates');
    this.destinationRoot('.');
    this.appName = 'myApp';
  },

  configFiles: function() {
    this.copy('_editorconfig', '.editorconfig');
    this.dir('code-analysis', '.');
    this.dir('pkg-conf', '.', {name: 'demo'});
  },

  appModule: function() {
    this.template('module/module.js_',
      'src/client/app/app.module.js');
    this.template('controller/ctrl.js_',
      'src/client/app.controller.js',
      {name: 'MyAppCtrl', appName: 'myApp'});
  }
});
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Yeoman runs npm & bower install for you
- Spawn CLI commands
- Execute in 'install' or 'end' running context

```
_skippedInstl: function() {
  if (this.options['skip-install']) {
   this.log('Run npm install & bower install');
  } else {
   this.spawnCommand('grunt', ['wiredep']);
  }
},
```

```
end: function() {
  this.installDependencies({
    skipInstall: this.options['skip-install'],
    skipMessage: this.options['skip-message'],
    callback: this._skippedInstl.bind(this)
  });
}
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Yeoman runs npm & bower install for you
- Spawn CLI commands
- Execute in 'install' or 'end' running context

```
_skippedInstl: function() {
  if (this.options['skip-install']) {
   this.log('Run npm install & bower install');
  } else {
   this.spawnCommand('grunt', ['wiredep']);
  }
},

end: function() {
  this.installDependencies({
    skipInstall: this.options['skip-install'],
    skipMessage: this.options['skip-message'],
    callback: this._skippedInstl.bind(this)
  });
}
```

Initialization

Interaction

Configuration

Generation

Installation

End

- Yeoman runs npm & bower install for you
- Spawn CLI commands
- Execute in 'install' or 'end' running context

```
_skippedInstl: function() {
  if (this.options['skip-install']) {
   this.log('Run npm install & bower install');
  } else {
   this.spawnCommand('grunt', ['wiredep']);
  }
},
```

```
end: function() {
  this.installDependencies({
    skipInstall: this.options['skip-install'],
    skipMessage: this.options['skip-message'],
    callback: this._skippedInstl.bind(this)
  });
}
```

Initialization

Interaction

Configuration

Generation

Installation

End

Short summary of the result

Next manual steps

(i.e. on –skip-installation)

After running `npm install` & `bower install`, inject your front end dependencies into your source code by running: grunt wiredep

Initialization

Interaction

Configuration

Generation

Installation

End

Short summary of the result

Next manual steps

(i.e. on –skip-installation)

After running `npm install` & `bower install`, inject your front end dependencies into your source code by running: grunt wiredep

Unit Tests

- Put each tested type of generator into separate describe block
- Mock user interaction and run generator in before block
- Test assertions in the it block

```
describe('angular generator', function() {
  before(function(done) {
    // run tested generator
    done();
  });
  describe('should generate following files', function() {
     it('bower.json', function() {
        // assert the file exits
        // assert correct file content
     });
     it('package.json', function() {

     });
  });
});
```

Unit Tests

- Put each tested type of generator into separate describe block
- Mock user interaction and run generator in before block
- Test assertions in the it block

```
describe('angular generator', function() {
  before(function(done) {
    // run tested generator
    done();
  });
  describe('should generate following files', function() {
     it('bower.json', function() {
        // assert the file exits
        // assert correct file content
     });
     it('package.json', function() {

     });
  });
});
```

Unit Tests

- Put each tested type of generator into separate describe block
- Mock user interaction and run generator in before block
- Test assertions in the it block

```
describe('angular generator', function() {
  before(function(done) {
    // run tested generator
    done();
  });
  describe('should generate following files', function() {
     it('bower.json', function() {
        // assert the file exits
        // assert correct file content
     });
     it('package.json', function() {

     });
  });
});
```

Unit Tests

```
describe('angular generator', function() {
  before(function(done) {
    helpers.run(path.join(__dirname, '../generators/app')
      .inDir(path.join(__dirname, './temp'), function(dir) {
       })
      .withArguments([]),
      .withOptions({'coffee': false}),
      .withPrompts({'bootstrap': true, 'routing': 'uiRouter'})
      .on('ready', function(generator) {
         generator.on('start', yoOutput.mute);
      })
      .on('end', function() {
         yoOutput.unmute();
         done();
      });
  });

  describe('should generate following files', function() {
    // test the assertions
  });
});
```

Unit Tests

```
describe('angular generator', function() {
  before(function(done) {
    helpers.run(path.join(__dirname, '../generators/app')
      .inDir(path.join(__dirname, './temp'), function(dir) {
       })
      .withArguments([]),
      .withOptions({'coffee': false}),
      .withPrompts({'bootstrap': true, 'routing': 'uiRouter'})
      .on('ready', function(generator) {
         generator.on('start', yoOutput.mute);
      })
      .on('end', function() {
         yoOutput.unmute();
         done();
      });
  });

  describe('should generate following files', function() {
    // test the assertions
  });
});
```

Unit Tests

```
describe('angular generator', function() {
  before(function(done) {
    helpers.run(path.join(__dirname, '../generators/app')
      .inDir(path.join(__dirname, './temp'), function(dir) {
       })
      .withArguments([]),
      .withOptions({'coffee': false}),
      .withPrompts({'bootstrap': true, 'routing': 'uiRouter'})
      .on('ready', function(generator) {
         generator.on('start', yoOutput.mute);
      })
      .on('end', function() {
         yoOutput.unmute();
         done();
      });
  });

  describe('should generate following files', function() {
    // test the assertions
  });
});
```

Unit Tests

```
describe('angular generator', function() {
  before(function(done) {
    // run tested generator
    done();
  });
  describe('should generate following files', function() {
     it('bower.json', function() {

        // assert the file exits
        assert.file('bower.json');

        // assert correct file content
        assert.fileContent('bower.json', /bootstrap/);
        assert.noFileContent('bower.json', /angular-route/);
     });

     it('package.json', function() {

     });
  });
});
```

Unit Tests

```
describe('angular generator', function() {
  before(function(done) {
    // run tested generator
    done();
  });
  describe('should generate following files', function() {
     it('bower.json', function() {

        // assert the file exits
        assert.file('bower.json');

        // assert correct file content
        assert.fileContent('bower.json', /bootstrap/);
        assert.noFileContent('bower.json', /angular-route/);
     });

     it('package.json', function() {

     });
  });
});
```

Modularization

- Subgenerators

```
this.composeWith('angular:controller', {
  arguments: ['admin'],
  options: {'matchingView': false}
  }
};
```

- External generators

```
this.composeWith('karma', {}, {
    local: require.resolve('generator-karma')
  }
};
```

- Shared templates and utilities

Modularization

- Subgenerators

```
this.composeWith('angular:controller', {
  arguments: ['admin'],
  options: {'matchingView': false}
  }
};
```

- External generators

```
this.composeWith('karma', {}, {
    local: require.resolve('generator-karma')
  }
};
```

- Shared templates and utilities

Modularization

- Subgenerators

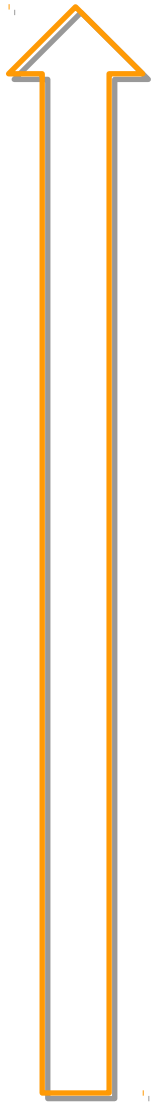
```
this.composeWith('angular:controller', {
  arguments: ['admin'],
  options: {'matchingView': false}
  }
};
```

- External generators

```
this.composeWith('karma', {}, {
    local: require.resolve('generator-karma')
  }
};
```

- Shared templates and utilities 

Scaffolding in Model-Driven Architecture



- Enterprise IT, BPM, B2B
- DevOps
- API, WS
- Language, DSL
- Model-Driven Development
- Runtime Code Generation
- **Scaffold**
- Project Seed
- Framework, Platform, SDK
- Runtime Module, Component, Library
- Copy-Paste
- Hand-Written Code

Pitfalls

- Asynchronous callbacks in conditional execution
- Post-write to files
- OS-specific path
- Naming
- Simplicity vs rich configurability

Questions

Scaffolding in JavaScript
Yeoman code generator

tomi vanek
software architect