AUTOMATING STYLE GUIDE DOCUMENTATION

zalando

FERIT TOPCU

@Fokusman

@Fokusferit

zalando

ZALANDO

- Europe’s leading online fashion platform founded 2008 in Berlin

>14K	15
Employees	Markets
~2000	>250K
Tech	Items

zalando

WHY A STYLE GUIDE?

THE SAME LOOK AND FEEL

DON’T REPEAT YOURSELF

MOVE FAST

HOW TO CONTRIBUTE?

EASY TO MAINTAIN

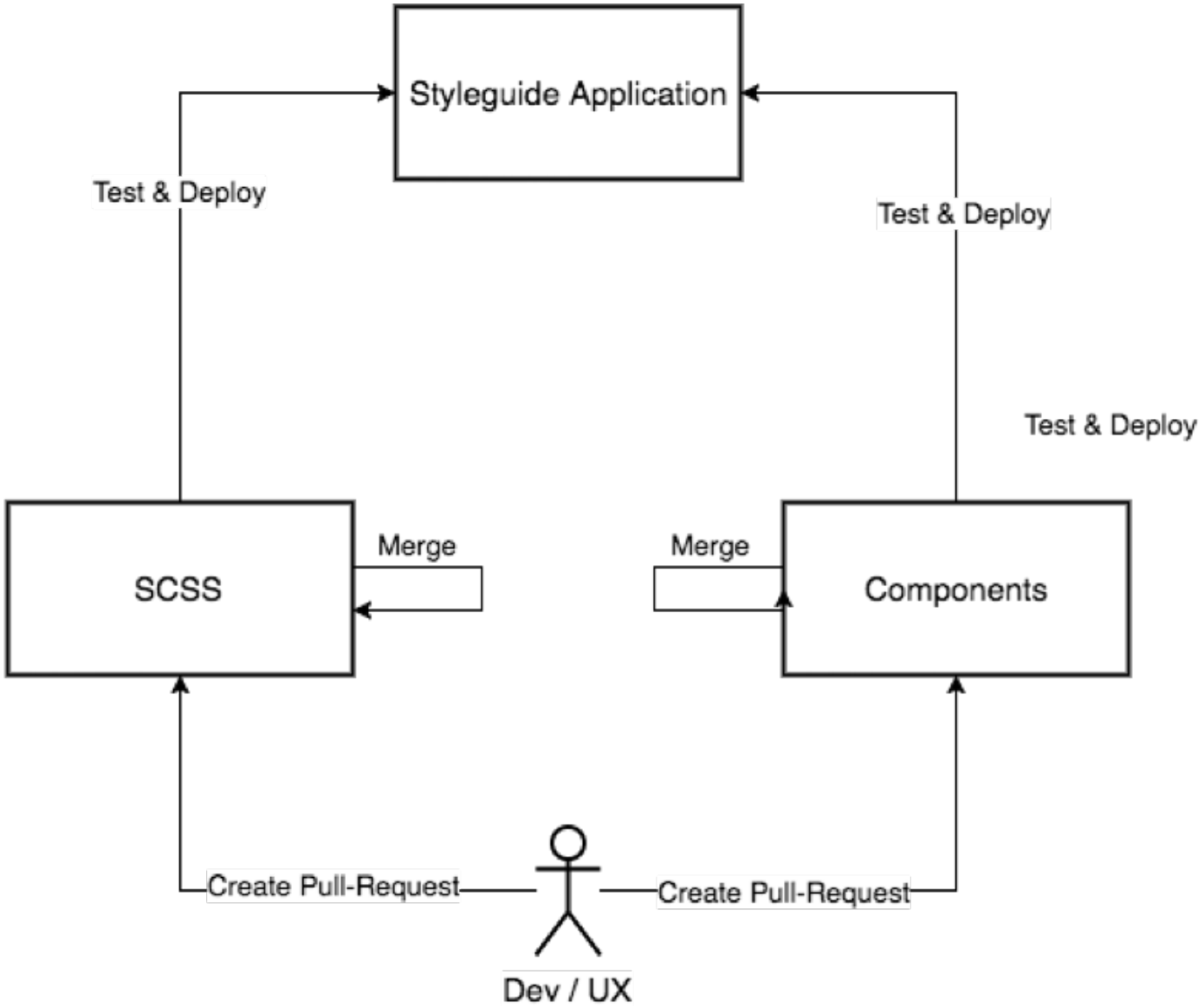
EASY TO CONTRIBUTE

EASY TO CUSTOMIZE

MARKDOWN


SOLUTION

WORKFLOW



zalando

TECHNICAL DETAILS (SCSS REPOSITORY)

```
"stylelint": {
  "extends": "stylelint-config-sass-guidelines",
  "plugins": [
    "stylelint-scss"
  ],
  "rules": {
    "max-nesting-depth": 2,
    "order/properties-alphabetical-order": null,
    "selector-no-qualifying-type": [
      true,
      {
        "ignore": [
          "attribute",
          "class"
        ]
      }
    ]
  }
}
}
```

```
const gulp = require('gulp');
const sass = require('gulp-sass');
const gulpStylelint = require('gulp-stylelint');

gulp.task('lint-sass', function() {
  return gulp.src(['./*.scss', './*/*.scss', '!./utils/*.scss'])
  .pipe(gulpStylelint({
    reporters: [
      {formatter: stylelintFormatter, console: true, fix: true}
    ]
  }));
});

// This is only used to validate the code is able to compile
gulp.task('sass', () =>
  gulp.src('./index.scss')
    .pipe(sass().on('error', sass.logError))
);

gulp.task('default', () => gulp.parallel('sass', 'lint-sass'));
```

TECHNICAL DETAILS (COMPONENTS)

```
import {React, Component, PropTypes} from '../imports';
import {CookieStorage} from './storage/cookie-storage';
import {LocalStorage} from './storage/local-storage';
import {Authorization} from './authorization';
import {WSDropdown} from '../ws-dropdown/ws-dropdown';

/**
 * This component renders a generic header which provides authentication and language management
 *
 * Optionally call WSHeader.setStorageType('cookie', 'zalando') If you want a to use cookies instead o
 * to persist the tokens. You can call WSHeader.getAccessToken().then(token => ...) to get the current
 * It will resolve null when no access token is present and therefore the user isn't logged in.
 * @property {Object} props React properties object
 * @property {string} props.loginUrl property used to set the application login url
 * @property {string} props.businessPartnerId property used to set the application businessPartnerId
 * @property {string} props.clientId property used to set the application yourturn clientId
 * @property {Array<Object>} props.links property used to set the list of links with multiple levels
 * @property {string} props.appName property used to set the application name
 * @property {string} props.appLogo property used to set the application image logo
 * @property {string} props.rootUrl property used to set the root application url
 * @property {Boolean} props.showLocale Flag used to show the locale dropdown
 * @property {Boolean} props.showAuthorization Flag to show the login area
 * @property {Function} props.onLocaleChange Function used to propagate data
 * @property {Function} props.onAuthChange Function used to propagate data
 */
export class WSHeader extends Component {
```

- Frontend Stack:
  - Webpack, Gulp
  - Babel
  - React
- Testing strategy:
  - React
  - Preact
  - Karma & Jasmine
- Documentation:
  - JSDocs
  - documentationjs

GENERATING COMPONENT DOCUMENTATION

```
/**
 * This component is rendering post mail adresses
 * @property {Object} address Address.property
 * @property {String} address.name Name of person / company
 * @property {String} address.street name of street to send with number
 * @property {Number} address.zip zip code
 * @property {String} address.city name of city
 *
 * @example
 *   render() {
 *     var testAddress = {
 *       name: 'Ferit Topcu',
 *       street: 'BerlinerStr.11',
 *       city: 'Berlin',
 *       zip: 10111
 *     };
 *     return (<AdressLine address={testAddress} />)
 *   }
 *
 */
export class AddressLine extends Component {

  /**
```

GENERATING COMPONENT DOCUMENTATION

```
/**
 * This component is rendering post mail adresses
 * @property {Object} address Address.property
 * @property {String} address.name Name of person / company
 * @property {String} address.street name of street to send with number
 * @property {Number} address.zip zip code
 * @property {String} address.city name of city
 *
 * @example
 *   render() {
 *     var testAddress = {
 *       name: 'Ferit Topcu',
 *       street: 'BerlinerStr.11',
 *       city: 'Berlin',
 *       zip: 10111
 *     };
 *     return (<AdressLine address={testAddress} />)
 *   }
 *
 */
export class AddressLine extends Component {

  /**
```

Parameters

- props

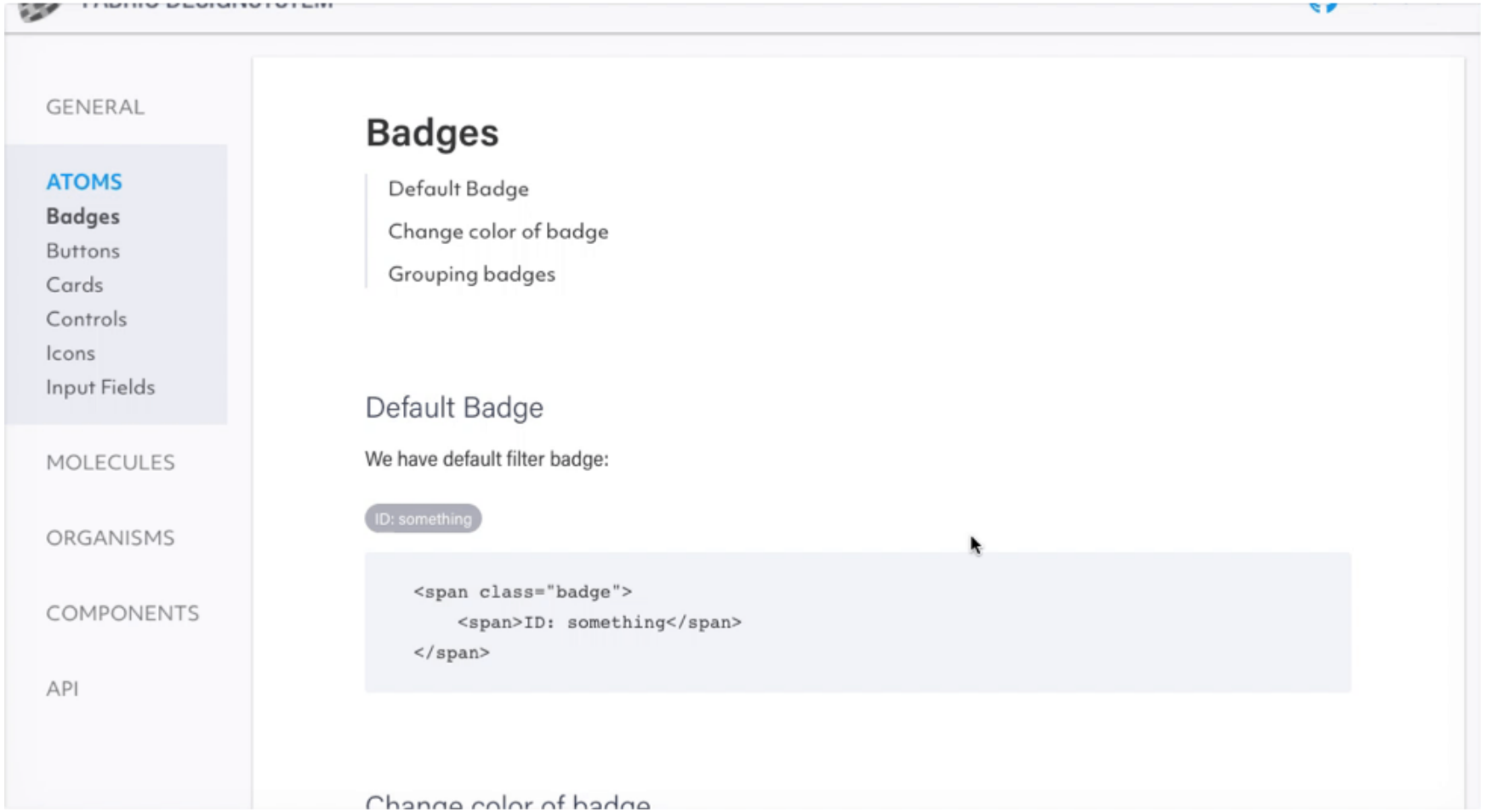
Properties

- address Object Address.property
  - address.name String Name of person / company
  - address.street String name of street to send with number
  - address.zip Number zip code
  - address.city String name of city

Examples

```
render() {
  var testAddress = {
   name: 'Ferit Topcu',
   street: 'BerlinerStr.11',
   city: 'Berlin',
   zip: 10111
  };
  return (<AdressLine address={testAddress} />)
}
```

AUTOMATED COMPONENT DOCUMENTATION



zalando

CONCLUSION

DOCUMENTATION AS IMPORTANT AS CODE

KEEP ENTRY LEVEL LOW

USE DOCUMENTATIONJS

JAMSTACK IS A GREAT APPROACH

https://github.com/fokusferit/documentationjs-demo-project

https://github.com/documentationjs/documentation

https://fabric-design.github.io/styleguide/


THANK YOU