

Recommended ESLint Rules for Enterprise-Level Angular Applications

For an enterprise-level Angular application, you'll want ESLint rules that enforce code quality, maintainability, and consistency across large teams. Here's a comprehensive set of recommendations:

Core Configuration

1. **Start with Angular ESLint:**

```
npm install --save-dev @angular-eslint/eslint-plugin @angular-eslint/eslint-plugin-template @typescript-eslint/eslint-plugin @typescript-eslint/parser eslint
```

2. **Recommended presets:**

```
"extends": [  
  "plugin:@angular-eslint/recommended",  
  "plugin:@angular-eslint/template/process-inline-templates",  
  "plugin:@typescript-eslint/recommended"  
]
```

Essential Rules for Angular

Component/Directive Rules

```
"@angular-eslint/component-class-suffix": ["error", {"suffixes": ["Component", "Page", "View"]}],  
"@angular-eslint/directive-class-suffix": ["error", {"suffixes": ["Directive"]}],  
"@angular-eslint/no-input-rename": "error",  
"@angular-eslint/no-output-rename": "error",  
"@angular-eslint/no-output-on-prefix": "error",  
"@angular-eslint/use-component-selector": ["error", {"type": "element", "prefix": "app"}],  
"@angular-eslint/use-injectable-provided-in": "error",  
"@angular-eslint/no-lifecycle-call": "error"
```

Template Rules

```
"@angular-eslint/template/banana-in-box": "error",
```

```
"@angular-eslint/template/no-negated-async": "error",
"@angular-eslint/template/cyclomatic-complexity": ["error", {"maxComplexity":
5}],
"@angular-eslint/template/no-call-expression": "error",
"@angular-eslint/template/eqeqeq": "error"
```

TypeScript Rules

```
"@typescript-eslint/explicit-member-accessibility": ["error", {"accessibility":
"explicit"}],
"@typescript-eslint/member-ordering": ["error", {
  "default": [
    "public-static-field",
    "private-static-field",
    "public-instance-field",
    "private-instance-field",
    "constructor",
    "public-static-method",
    "private-static-method",
    "public-instance-method",
    "private-instance-method"
  ]
}],
"@typescript-eslint/no-inferable-types": ["error", {"ignoreParameters": true}],
"@typescript-eslint/no-unused-vars": ["error", {"args": "none"}],
"@typescript-eslint/consistent-type-definitions": ["error", "interface"],
"@typescript-eslint/explicit-function-return-type": ["error", {"allowExpressions":
true}],
"@typescript-eslint/naming-convention": [
  "error",
  {
    "selector": "interface",
    "format": ["PascalCase"],
    "prefix": ["I"]
  },
  {
    "selector": "typeAlias",
    "format": ["PascalCase"],
    "prefix": ["T"]
  }
]
```

]

General Best Practices

```
"curly": "error",
"eqeqeq": ["error", "always", {"null": "ignore"}],
"no-console": ["error", {"allow": ["warn", "error"]}],
"no-empty": ["error", {"allowEmptyCatch": true}],
"no-eval": "error",
"no-implied-eval": "error",
"no-magic-numbers": ["error", {"ignore": [-1, 0, 1]}],
"no-return-await": "error",
"no-throw-literal": "error",
"no-unused-expressions": "error",
"prefer-const": "error",
"radix": "error",
"require-await": "error"
```

Enterprise-Specific Additions

1. **Documentation Requirements:**

```
"jsdoc/require-jsdoc": ["error", {
  "require": {
    "FunctionDeclaration": true,
    "MethodDefinition": true,
    "ClassDeclaration": true
  }
}]
```

1. **Complexity Control:**

```
"complexity": ["error", 10],
"max-depth": ["error", 4],
"max-lines-per-function": ["error", 50],
"max-params": ["error", 4]
```

1. **Security Rules:**

```
"security/detect-bidi-characters": "error",  
"security/detect-non-literal-fs-filename": "error",  
"security/detect-possible-timing-attacks": "error"
```

Setup Recommendations

1. Create separate ESLint configs for:
 - Main application code
 - Unit tests (more relaxed rules)
 - E2E tests
 2. Use .eslintignore to exclude generated files and third-party code
 3. Add pre-commit hooks with husky and lint-staged to enforce linting
 4. Integrate with your CI pipeline to fail builds on lint errors
 5. Consider using eslint-plugin-deprecation to catch deprecated APIs
- Would you like me to elaborate on any specific area of these recommendations?