Съдържание

1. done - Какво е linter?
2. done - Представяне на JavaScript Lint
3. done - Какво открива JavaScript Lint?
4. done - Начини на използване
5. done - Конфигуриране
6. Опции
7. Един пример
8. Плюсове и минуси на JavaScript Lint
9. Сравнение с други linter-и за JavaScript
10. Заключение
11. Литературни източници
12. Списък с фигури
13. Списък с примерен програмен код

Текст

1. Какво е linter?
   1. Linter-ите представляват софтуерни инструменти, които извършват анализ на изходния код на програмата и помагат за откриване на някои често срещани грешки и проблеми. Те също така могат да бъдат използвани и за прилагане на конкретни стилистични правила за писане на код, наложени от дадена организацията (style guides).
   2. Използването на linter помага бързо да се справим с потенциален проблем, дори преди той да се проявил като грешка или изключение по време на изпълнение на програмата. Възможно грешката да е елементарна като използване на недекларирана променлива, но ако трябва да дебъгваме може да загубим изключително много време да намерим конкретната линия код с грешката.
   3. При изпълнението на инструмента откриването на проблем се автоматизира и ускорява значително. Той има дефинирани правила за езика, за който се използва и при откриването на нарушения генерира предупреждение кое място в кода какво конкретно правило нарушава. Някои инструменти има опция автоматично и да поправят грешките.
   4. Към подмножество на linter-ите попадат и инструментите, които само форматират кода спрямо някакви правила, т.е. правят го да изглежда по-красив. Пример за това е Prettier.
2. Представяне на JavaScript Lint
   1. JavaScript Lint е инструмент, предназначен за извършване на статичен анализ на JavaScript код. Неговата разработка е била активна в начало на века, като за последен път през 2007 година на официалния сайт има някакви новини или обновления.
   2. <картинка на сайта>
   3. Инструментът е безплатен, с отворен код. От официалния му сайт [1] е възможно да се изтегли готовия за работа инструмент или архив, съдържащ изходния код. Има линк към хранилище на Subversion, но не изглежда, че той не е активен към момента на писането.
   4. Платформите на които може да се използва са Windows, като не е специфицирана версията, но най-вероятно към времето на последна активност това е било XP или Vista. Другият вариант е Mac OS X, където също не е упомената версията на операционната система.
3. Какво открива JavaScript Lint
   1. Тъй като JavaScript е език за програмиране, който се интерпретира, слабо и динамично типизиран, то нуждата от инструменти, които да ни предпазват от грешки е задължителен. Нека разгледаме някои от най-често срещаните грешки при писане на JavaScript код, а именно тези, които linter-ите ни помагат бързо да открием и поправим.
      1. използване на променлива ли метод, който не е деклариран : предизвиква се неочаквано поведение на програмата. Такива грешки могат да се открият при наличие на достатъчно добри тестове, но с използването на lint инструменти идентифицирането се улеснява значително.
      2. променлива ли метод, който никога не се използва в кода : чрез откриването и отстраняването на този проблем се премахва ненужен ресурс
      3. блок от код, който никога не се изпълнява, заради return, continue, break или throw оператори : идентифицира се евентуален проблем със потока на изпълнението на програмата или ненужен фрагмент от код
      4. пропуснати ограждащи скоби на изразите if, for, while : това води до влошаване на четливостта на кода, особено ако на реда след оператора има и някакъв друг израз. Потенциално място за бъгове, както и създава допълнителна работа ако трябва в тялото след оператора да се добави повече от един израз
      5. липсващи break изрази при case случаите на switch оператор : това е потенциално място за бъгове свързани с потока на изпълнението на програмата.
      6. нарушаването на правила, записани в даден style guide : непостоянството в начина, по който пишем кода го прави нечетлив и изглежда неприятно. При налагане на единен стил, се подобрява четливостта и се създава чувството за колективно притежание но кода (code ownership), т.е. четейки код, който не е писан от теб, но изглежда стилистично като твоя.
4. Начини на използване

Съществуват няколко начина за интегрирането на JavaScript Lint в процеса ни на работа, като например

* 1. директно пускаме програмата, чрез нейния CLI : може би най-неудобния вариант, тъй като изхода, който ще получим няма да е толкова четим и интерактивността със самия код почти липсва. Въпреки това не се изискват никакви допълнителни настройки, а евентуално само указване на конфигурационен файл, от който да посочва за какви грешки да се търси. Подходът е приложим при работа с малки и средни проекти, където просто искаме да се уверим, че няма никакви сериозни грешки.
  2. използване от контекстното меню на Windows Explorer : в документацията на JavaScript Lint е посочен този вариант, като идеята е от Windows Explorer да кликнем с десен бутон върху файл с JavaScript код и така да активираме инструмента. Резултатите вероятно отново ще получим в терминала и отново сме в горния случай – ниска интерактивност и дори още по ниска практичност, защото трябва да повтаряме действието за всеки .js файл, който имаме в проекта.
  3. приставка/добавка в IDE или редактор : практичен и удобен вариант, като са налице някои предимства. Обикновено има обособена секция за изхода от инструмента. Също така при кликване върху намерените грешки, които са открити курсорът се отвежда на мястото на грешката, показване на потенциални решения на проблема и конфигуриране на инструмента от настройките на IDE-то. За съжаление поради спрялата разработка на JavaScript Lint не се поддържат съвременни IDE-та или редактори.
  4. Можем да отбележим и наличието на уеб вариант на инструмента на сайта му[1]. За съжаление установихме, че той не работи правилно и не дава никакъв изход.
  5. <картинка>
  6. В примера грешката е, че променливата sum не е дефинирана, а е използвана в пресмятането. Инструментът не показва и предпреждение за изпуснати скоби около for цикъла.

1. Конфигуриране и използване
   1. Ние ще използваме софтуера JavaScript Lint при версия 0.3.0 за Windows, използвайки файла jsl-0.3.0-win32.zip, изтеглен от сайта на инструмента[1]. Изпълняваме софтуера под ОС Windows 10.
   2. Грешките, за които търси могат да се конфигурират по два начина. Единият е използвайки вградените настройки, а вторият е наличие на конфигурационен файл, който да бъде подаден като параметър на програмата.
   3. JavaScript Lint предлага основно работа чрез конзолен интерфейс. Като основни параметри са път до конфигурационен файл и файлове с JavaScript код, които да бъдат прегледани.
   4. <картинка с опциите>
2. Опции

Опциите на инструмента се задават в конфигурационния файл, който се подава при изпълнение. В конфигурационния файл по подразбиране са представени всички възможни опции на JavaScript Lint, както и как да се активират. Нека разгледаме някои от полезните опции

1. Един пример

Ще направим един прост пример, за да покажем основните опции на JavaScript Lint. За целта ще използваме наш конфигурационен файл, който е копие на този по подразбиране с тази разлика, че са включени предупрежденията за използване на недеклариран идентификатор(променлива, метод), нуждата от ограждащи скоби около оператор и сме дефинирали console като глобален обект.

1. Плюсове и минуси на JavaScript Lint
   1. ul с li-та ?
   2. плюсове
      1. лек, няма dependencies
      2. подходящ при малки проекти
   3. минуси
      1. няма auto fix
      2. не бачка интегрирането с нищо
      3. не може да се приложи съвременен style guide
2. Сравнение с други linter-и за JavaScript
   1. таблица с плюсове и минуси
3. Заключение