Κανόνες που δεν θεωρείται οτι θα έδιναν αξία

Κανονες για Πιθανά Ζητήματα Λογικής

- constructor-super - Εχουμε φύγει από την λογική των κλάσεων σε React Components. Εφόσον πολύ σπάνια πλέον χρησιμοποιείται κλάση στο front δεν έχει πολύ αξία να δώσει.Σε ο,τι υπάρχει θεωρώ περισσότερο κακό παρά καλό θα έκανε να κοκκίνιζε και κάτι ακόμα
- no-class-assign - Οτι ειπώθηκε και για το παραπάνω
- no-constructor-return - Οτι ειπώθηκε και για τα παραπάνω
- no-dupe-class-members - Οτι ειπώθηκε και για τα παραπάνω. Αναφερεται μονο για classes.
- no-irregular-whitespace - Μπορεί να επιτευχθεί με prettier. Περαν αυτού και ενοχλητικό είναι να σου σκάει μια γραμμή επειδή εχασες ένα κενό αλλά και αν υπάρξει conflict prettier / eslint θα χανεται τζαμπα χρόνος στο φτιάξιμο.
- no-misleading-character-class - Δεν θεωρω οτι θα πρόσθετε αξία στο έργο ένας κανόνας που ασχολείται με emojis
- no-prototype-builtins - Δεν υπάρχουν πολλές περιπτώσεις τετοιου τυπου που να εχω δει επομένως δεν ξέρω και πόση αξία θα είχε να προστεθεί ένας κανόνας που δεν χρειάζεται
- no-sparse-arrays - Αν καποιος developer για τους δικούς του λόγους ήθελε να κάνει ένα sparse array ίσως να ήταν καλό να του δίνουμε την δυνατότητα. Εχω δει περιπτώσεις οπου χρησιμοποιείται το συμβολο _ αντί του κενού για περιπτωσεις όπου δεν μας νοιάζει η μεταβλητη. Θεωρω πιο καθαρό να μην υπήρχε ούτε αυτό και να υπήρχε ο αραιός πίνακας στην θεση του.
- no-this-before-super - Δεν εχει χρησιμότητα
- no-unmodified-loop-condition - Περισσότερο θα μπερδευε παρά θα ωφελούσε. Δεν χρησιμοποιείται καν πλέον.
- no-unsafe-negation - Περιέχει προβληματική συμπεριφορά και αυτό που πραγματεύεται σχετικά μικρό. Θα βοηθούσε το readability αλλά εφόσον υπάρχουν υποσημειώσεις σχετικά με αυτό καλύτερα να μην μπει
- no-unused-private-class-members - Δεν θα χρησιμοποιηθεί κατα πάσα πιθανότητα

Προτάσεις για Consistency Γραφής μέσω Κώδικα

- accessor-pairs - Θα μπερδευε καποιον που δεν το ξερει για ώρες, είναι αχρείαστο και ξεπερασμένο.
- class-methods-use-this - Αχρείαστο
- consistent-this - Αυτο χρησιμοποιείται τόσο σπάνια (αν χρησιμοποιείται) που αν χρησιμοποιείται ας έχει ότι descriptive όνομα σκεφτεί ο προγραμματιστής.
- func-names - Δεν θα εδινε σημαντικό value
- grouped-accessor-pairs - Αχρειαστο

- guard-for-in - Το loop for…in χρησιμοποιειται σπάνι στο frontend. Ο έλεγχος που προτείνει ίσως να ήταν καλός για καποιον που δεν ξερει οτι το object εχει και inherited properties από το Prototype. Ωστόσο ακομα και αν καποιος το ηξερε παλι ίσως τον μπερδευε περισσότερο το μηνυμα του λιντερ παρα αυτο που θα ήθελε να κάνει. Οποτε ειναι κατι που ασχολείται με ένα σπάνιο φαινόμενο.
- max-classes-per-file - Δεν χρησιμοποιείται στο εργο
- no-caller - Δεν θα προσδώσει αξία. Απλά θα δυσκολέψει τη ζωή καποιου που όντως θέλει να χρησιμοποιήσει τα arguments.
- no-case-declarations - Πολύ σπάνια χρησιμοποιήσιμο και πολύ συγκεκριμένος ο κανόνας που πραγματεύεται χωρίς να χρειάζεται
- prefer-template - Δεν ενθαρρύνονται τα literals εναντι των υπολοίπων τρόπων καθώς και λιγότερο performant είναι (μικρό το κακό) αλλά μετατρέπουν αυτόματα και με πολύ συγκεκριμένο τρόπο strings τα οποια ίσως να μην βοηθούν. Το πρόβλημα που λύνουν κρίνεται λιγότερο σημαντικό από την λύση.
- radix - Περιττός για το συγκεκριμένο εργο
- prefer-regex-literals - Υπάρχουν περιπτώσεις όπου είτε το constructor(String) είτε το regExp έχουν το πλεονέκτημα ειδικά αν η επικοινωνία γίνεται με BE. Καλυτερα να εχουμε ενα flexibility και να μην ενθαρρύνουμε κάποιον από τους 2 τρόπους.
- prefer-numeric-literals - Πολύ ειδικό σενάριο που δεν ξερω αν θα βοηθησει η θα μπερδεψει.
- prefer-numeric-literals - Επίσης ειδικό σενάριο. Δεν βλεπω σημαντική αξια ωστε να εμποδίσουμε την χρηση του parseInt κατα αυτο τον τρόπο.
- prefer-named-capture-group - Έχουν εντοπιστεί προβλήματα οπότε καλύτερα να μην χρησιμοποιουνταν ο κανόνας αυτός.
- prefer-exponentiation-operator - Ο x**y εναντι του pow(x,y). Θεωρω οτι το να κρατησουμε flexibility στο συγκεκριμενο δεν θα ήταν κακή ιδέα. Και τα δύο είναι γνωστοί τρόποι οπότε ο καθένας ας γραφει με αυτό που νομίζει καλυτερο.
- no-with - Ισως χρήσιμο αλλά πιθανότατα δεν χρησιμοποιείται πλέον
- no-warning-comments - Πιθανόν χρήσιμο αλλά υπάρχουν και σημαντικότερα ζητήματα να ασχοληθούμε.
- no-useless-rename - Χρήσιμο αλλά μαλλον ελέγχεται ήδη οταν γίνεται compiling.
- no-useless-constructor - Αχρείαστο
- no-ternary - Αν χρησιμοποιείται σωστά ο ternary operator είναι σημαντικό εργαλείο για καλύτερο readability κωδικά. Καταχρήσεις του δεν θα πρέπει να οδηγούν στην απαγόρευση του.
- no-restricted-globals - Περιττό να προστεθεί
- no-restricted-exports - Περιττό
- no-restricted-imports - Ίσως χρήσιμο για ορισμένες περιπτώσεις αρχείων έτσι ώστε να καταλαβαινει οτι το αρχείο που θελουμε να κάνουμε import δεν είναι σωστό να γίνει import. Αλλά μαλλον αχρηστος κανονας για το εργο
- no-regex-spaces - Φαίνεται να υπάρχουν προβληματα με τον κανόνα αυτό. Φαινεται ήσσονος σημασίας
- no-redeclare - Χρησιμο αλλά αν υπάρχει ήδη είναι αχρειαστο
- no-proto - Αχρειαστο

- no-plusplus - Κανόνας που θα δυσκολέψει το development αν ακολουθηθεί χωρίς να δώσει κάτι σημαντικό
- no-param-reassign - Περιττός γιατί εχουμε strict mode λόγω webpack
- no-octal - Δεν δινει αξια στο εργο. Ελαχιστα οκταδικά αλλά και οκταδικό να προστεθεί δεν είναι τραγικό πρόβλημα
- no-nonoctal-decimal-escape - Αχρείαστος
- no-nested-ternary - Μια καλή μορφοποιηση μεσω prettier θα βοηθούσε στην κατανοηση του κώδικα και ταυτοχρονα θα είχαμε το flexibility εξετασης παραπάνω από μιας συνθηκης μεσω του nested ternary.
- no-multi-str - Καλύτερα να εχουμε το flexibility του multiline string καθως εχει χρειαστεί πολλές φορές να παρσαρω επεξεργαστω τετοια ήδη string. Επίσης ίσως να εχει conflicts και με το prettier κατι που θα μειωνε το development experience.Τελος η προσθηκη του warning θα δυσκόλευε την ανάλυση ενός τέτοιου string καθως αποσυντονίζει. Θα μπορουσαμε να βαλουμε ενα καλό κανονα στο formatting και να τελειώναμε.
- no-label-var - Αχρειαστο
- no-implied-eval - Ο κανόνας θα ταίριαζε πιο πολύ σε Node project παρα στο δικό μας
- no-global-assign - Δεν υπαρχει μεγαλη αναγκη για τον συγκεκριμένο.
- no-extra-label - Αχρειαστο
- no-extra-bind - Αχρείαστος καθως δεν χρησιμοποιείται το bind στο front πλέον. Εκτός και αν γραφτεί καμια κλάση πουθενα.
- no-eq-null - Αχρείαστος για αυτό ο κανόνας eqeqeq επιβαλει και τον συγκεκριμένο κανόνα
- no-div-regex - δεν βλεπω πολύ μεγάλη αξία. Πιο πολύ θα μπερδευε χωρίς να φτιαχνει κατι λάθος
- yoda - Με τα εργαλεία που έχουμε πλέον δεν χρειάζεται.
- vars-on-top - Χρησιμος κανόνας αλλά για άλλες προηγούμενες εκδόσεις
- strict - Περιττός κανόνας
- require-await - Δεν θεωρω οτι θα βοηθούσε ο κανόνας στο front ιδιαίτερα.