TSLint   Usage   Rules   Formatters   Develop   News

Rule: curly

Enforces braces for `if` / `for` / `do` / `while` statements.

Rationale

```
if (foo === bar)
    foo++;
    bar++;
```

In the code above, the author almost certainly meant for both `foo++` and `bar++` to be executed only if `foo === bar`. However, they forgot braces and `bar++` will be executed no matter what. This rule could prevent such a mistake.

Notes:

🔧 **Has Fixer**

Config

One of the following options may be provided:

- `"as-needed"` forbids any unnecessary curly braces.
- `"ignore-same-line"` skips checking braces for control-flow statements that are on one line and start on the same line as their control-flow keyword

Config examples

```
"curly": true
```

```
"curly": [true, "ignore-same-line"]
```

```
"curly": [true, "as-needed"]
```

Schema

```
{
  "type": "array",
  "items": {
    "type": "string",
    "enum": [
      "as-needed",
      "ignore-same-line"
    ]
  }
}
```

Code examples:

Require curly braces whenever possible (default)

```
"rules": { "curly": true }
```

Passes

```
if (x > 0) {
    doStuff();
}
```

Fails

```
if (x > 0)
    doStuff();

if (x > 0) doStuff();
```

Make an exception for single-line instances

```
"rules": { "curly": [true, "ignore-same-line"] }
```

Passes

```
if (x > 0) doStuff();
```

Fails

```
if (x > 0)
    doStuff()
```

Error on unnecessary curly braces

```
"rules": { "curly": [true, "as-needed"] }
```

Passes

```
if (x > 0)
    doStuff();

if (x > 0) {
    customerUpdates.push(getInfo(customerId));
    return customerUpdates;
}
```

Fails

```
if (x > 0) {
    doStuff();
}
```

 palantir/tslint

 PalantirTech