

Simplifying Proxies with Reflect



Shawn Wildermuth

Instructor, Microsoft MVP, and Filmmaker

@shawnwildermuth | shawn.wildermuth.com



Overview



Reflect API

- What is Reflect
- How Reflect can simplify Proxy objects
- How a complete example of Proxy looks





What is Reflect?

- Static Object
- API for performing proxy operations
- Does not require a proxy
- Mirrors the Handler



Demo



Using Reflect



Refresher – What is a Proxy

Proxy

customer

```
props: Array  
[  
  firstName,  
  lastName,  
  formatName  
]
```

Handler (Façade)

```
get {},  
set {},  
apply {},  
defineProperty {},  
construct {}
```



Rules for Handler

Proxy

customer

props: Array

[

firstName,

lastName,

formatName

]

Handler (Façade)

get {},

set {},

apply {},

defineProperty {},

construct {}

Returns value

Returns Boolean

Returns result

Returns Boolean

Returns object




```
const handler = {  
  get() {...},  
  set() {...},  
  apply() {...},  
  has() {...},  
  defineProperty() {...},  
  deleteProperty() {...},  
  construct() {...},  
  ...  
};
```

```
Reflect.get(...);  
Reflect.set(...);  
Reflect.apply(...);  
Reflect.has(...);  
Reflect.defineProperty(...);  
Reflect.deleteProperty(...);  
Reflect.construct(...);  
...
```

◀ **Handler supports many operations**

◀ **Reflect has same operations**



Demo



Using Reflect in a Proxy



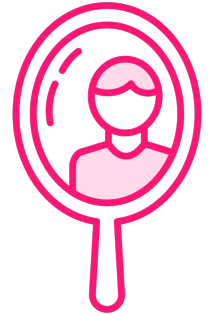
Demo



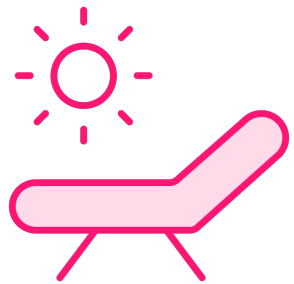
Complete Example of Proxy



Where are we?



Reflect object can simplify some Proxy handlers



While Reflect can be used outside a Proxy, it's very rare



The more your Proxy does, the slower it will run

