DR INŻ. PIOTR SIEWNIAK

# Moduły lokalne

## Eksport/import modułów



Można wyeksportować albo wszystkie, albo wybrane zasoby danego pliku (modułu), np. wybrane funkcje, wybrane obiekty itd. Wyeksportowane zasoby danego modułu stanowią wówczas jego zasoby publiczne – czyli takie, które można wykorzystać w aplikacjach. Zasoby prywatne modułu – czyli takie, które nie zostały wyeksportowane – nie są dostępne na zewnątrz modułu.

Moduły NODE można eksportować na dwa sposoby, przy wykorzystaniu:

1. obiektu `exports`;
2. właściwości `exports` obiektu `module`.

Używając obiektu `exports` można wyeksportować dowolną liczbę zasobów danego modułu, np. wybrane funkcje, wybrane obiekty, klasy itp.

Przykład eksportu dwóch niezależnych funkcji: `polePr()` i `obwodPr()` oraz klasy `Pomieszczenie`:

```
exports.polePr = polePr;
exports.obwodPr = obwodPr;
exports.Pomieszczenie = Pomieszczenie;
```

Dobrym nawykiem programistycznym jest, ażeby instrukcje eksportu zasobów publicznych modułu znajdowały się na końcu jego kodu źródłowego.

Oprócz rozwiązania przedstawionego powyżej można eksportować zasoby na bieżąco – tj. w wyrażeniach zawierających ich definicje, np.:

```
exports.polePr = function(b1, b2) {
        return b1 * b2;
}
exports.obwodPr = (b1, b2) => {
        return 2 * b1 + 2 * b2;
}
```

Składnia użycia właściwości `exports` obiektu `module` jest analogiczna do składni obiektu `exports`, np.:

```
module.exports.polePr = function(b1, b2) {
        return b1 * b2;
}
module.exports.obwodPr = (b1, b2) => {
```

```
        return 2 * b1 + 2 * b2;
}
```

Jak wspomniano już wcześniej, instrukcje eksportu zasobów publicznych modułu powinny znajdować się na końcu jego kodu źródłowego, np.:

```
// Eksport obiektu personLiteral i funkcji-konstruktora personConstructor():
module.exports.person = personLiteral;
module.exports.Person = personConstructor;
```

Obiekt `PersonLiteral` został wyeksportowany jako właściwość `person` obiektu `module.exports`. Funkcja (konstruktor) `personConstructor` został wyeksportowany jako właściwość `Person` obiektu `module.exports`.

```
// Eksport obiektu pracownik1, konstruktora Pracownik2() i klasy Pracownik3:
module.exports = {
   pracownik1,
   Pracownik2,
   Pracownik3
}
```

**UWAGA**

`Module.exports` oraz `exports` wskazują na ten sam obiekt (dokładniej: stanowią referencje do tego samego obiektu), który na początku jest pusty. Dlatego też, nie ma znaczenia, która z tych konstrukcji zostanie wykorzystana w celu eksportu zasobów.

Jednakże, należy zwrócić uwagę na to, że jeśli w jednym module zostaną użyte równocześnie obie wymienione powyżej konstrukcje oraz określone zasoby zostaną przypisane bezpośrednio do `module.exports`, wówczas wyeksportowane zostaną wyłącznie te zasoby. Wywołania instrukcji `exports` zostaną w tym przypadku zignorowane.

Import (dołączenie) zasobów publicznych do aplikacji jest realizowane za pomocą funkcji standardowej `require()`. Funkcja ta zwraca obiekt, zawierający te zasoby. Np. wywołanie

```
const f = require('./custom_modules/difference');
```

zwraca obiekt zawierający zasoby publiczne modułu `difference` z katalogu `custom_modules`.