

PHP: creare e usare le interfacce

In PHP l'uso delle interfacce è controllato da regole specifiche.

Definiamo un'interfaccia che possieda un metodo per generare un token di autenticazione:

```
interface Security {  
    public function createToken();  
}
```

A questo punto implementiamo il metodo specificato in due classi distinte. Il metodo implementato deve avere lo stesso nome, la stessa visibilità e la stessa firma del metodo definito nell'interfaccia: questa è la regola fondamentale da seguire quando una classe implementa un'interfaccia.

```
class User implements Security {  
    protected $id;  
  
    public function __construct($id) {  
        $this->id = $id;  
    }  
  
    public function createToken() {  
        return uniqid();  
    }  
}  
  
class SessionManager implements Security {
```

```
private $session;

public function __construct() {
    $this->session = $_SESSION;
}

public function createToken() {
    return md5(uniqid() .
$_SERVER['HTTP_USER_AGENT']);
}
}
```

Come si può notare la differenza sta nell'implementazione scelta per il metodo. La documentazione di PHP infatti afferma:

Object interfaces allow you to create code which specifies which methods a class must implement, without having to define how these methods are handled.

Comprendere il funzionamento delle interfacce è il primo passo per sviluppare un OOP polimorfico in PHP.