

Difference between Service Locator and Dependency Injection?

The difference between **Service Locator** and **Dependency Injection** lies in how objects obtain their dependencies. In **Service Locator**, objects autonomously search for required dependencies in a specialized container, while in **Dependency Injection**, dependencies are passed to the object from the outside. Let's consider an example in **PHP**:

```
class ServiceLocator {
    private $services = [];

    public function __construct() {
        // Register services in the container
        $this->services['userService'] = function() {
            return new UserService();
        };
    }

    public function getService($serviceName) {
        if (isset($this->services[$serviceName])) {
            return $this->services[$serviceName]();
        }
        throw new Exception('Service not found.');
```

```

    }
}

class UserService {
    public function getUser() {
        return 'User';
    }
}

$serviceLocator = new ServiceLocator();
$userService = $serviceLocator->getService('userService');
echo $userService->getUser(); // Outputs: User
```

Example of Dependency Injection in PHP:

```
class UserService {
    public function getUser() {
        return 'User';
    }
}

class UserController {
    private $userService;

    public function __construct(UserService $userService) {
        $this->userService = $userService;
    }

    public function showUser() {
        echo $this->userService->getUser();
    }
}

$userService = new UserService();
$userController = new UserController($userService);
$userController->showUser(); // Outputs: User
```