



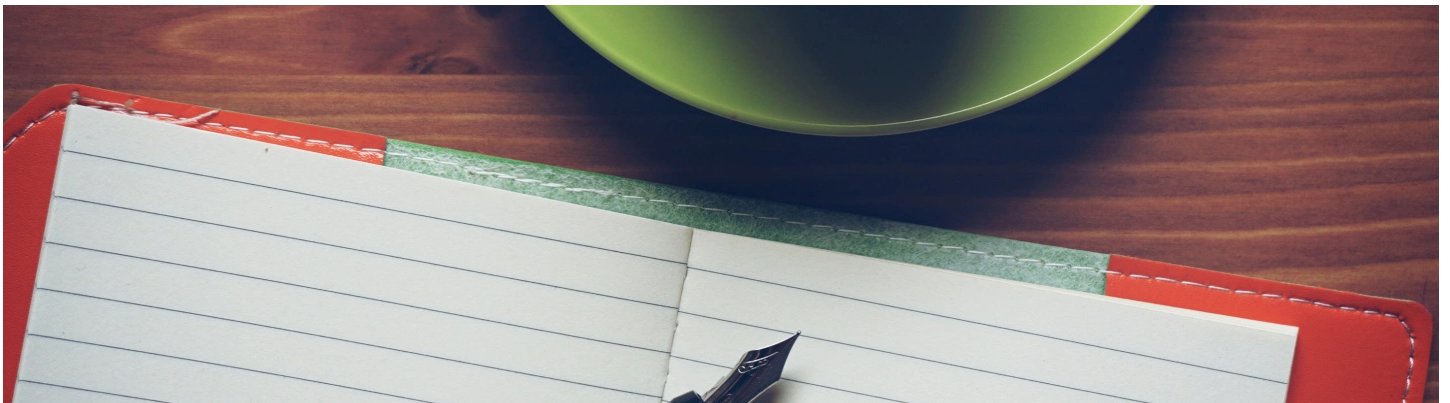
# devs4j

EL MEJOR SITIO WEB SOBRE PROGRAMACIÓN EN ESPAÑOL.

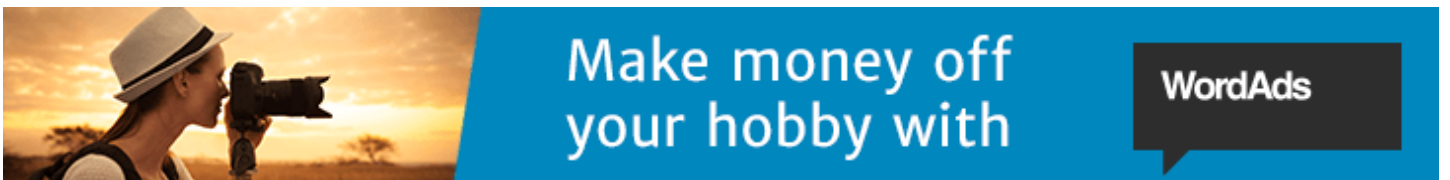
[HOME](#)

[ABOUT](#)

[CONTACT](#)



Anuncios



[REPORT THIS AD](#)

# Spring framework 5 : Uso de la anotación @Quali fier

 HACE 5 DÍAS     DEJA UN COMENTARIO

1 Vote

En el post anterior vimos como utilizar dependency injection [Spring framework 5 : Dependency Injection utilizando Spring](#) (<https://devs4j.com/?p=7492>) ahora es turno de analizar el uso de la anotación @Qualifier y como nos ayuda al momento de utilizar Spring framework.

## Definiendo los servicios de spring

En el post anterior vimos como funciona Dependency Injection y como utilizarlo entre diferentes Beans, ahora veamos el siguiente problema para entender el funcionamiento de la anotación @Qualifier :

```
/**
```

```
* @author raidentrance
```

```
*  
*/  
  
public interface UserService {  
  
    public boolean authenticate(String username  
  
}
```

Como vemos tenemos una interfaz llamada User service con el método authenticate el cual recibe username y password y devuelve un boolean indicando si se logro autenticar exitosamente.

```
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.stereotype.Service;  
  
/**  
 * @author raidentrance  
 *  
 */  
@Service  
public class UserServiceDatabaseImpl implements Use  
  
    private static Logger log = LoggerFactory.g  
  
    @Override  
    public boolean authenticate(String username  
        log.info("authenticating against th  
        return false;  
    }  
  
}
```

```
import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.stereotype.Service;

/**
 * @author raidentrance
 *
 */
@Service
public class UserServiceActiveDirectoryImpl implements

    private static Logger log = LoggerFactory.g

    @Override
    public boolean authenticate(String username
        log.info("authenticating against th
        return false;
    }

}
```

Ahora podemos ver que se tienen 2 implementaciones, una llamada **UserServiceDatabaseImpl.java** y la otra **UserServiceActiveDirectoryImpl.java** una puede autenticar utilizando una base de datos y la otra a través de active directory, para este ejemplo simplemente escribiremos en el log para no sobre complicar el ejemplo.

## Utilizando los servicios de spring

Una vez que tenemos los dos servicios definidos el siguiente paso será utilizarlos, veamos la siguiente implementación:

```
import org.springframework.beans.factory.annotation
import org.springframework.stereotype.Controller;

/**
 * @author raidentrance
 *
 */
@Controller
public class UserController {

    @Autowired
    private UserService userService;

    public boolean authenticate(String username
        return userService.authenticate(use
    }
}
```

Si ejecutamos la aplicación utilizando la implementación de UserController anterior tendremos el siguiente error:

```
Field userService in com.devs4j.qualifier.UserContr
- userServiceActiveDirectoryImpl: defined i
- userServiceDatabaseImpl: defined in file
```

Esto se debe a que utilizamos la interfaz como referencia al bean pero existen 2 implementaciones, como vemos se hace inyección por tipo así que spring no tiene forma de saber que bean se inyectará en el atributo userService;

Para resolver el problema anterior debemos modificar la clase

**UserController.java** como se muestra a continuación:

```
import org.springframework.beans.factory.annotation
import org.springframework.beans.factory.annotation
import org.springframework.stereotype.Controller;

/**
 * @author raidentrance
 *
 */
@Controller
public class UserController {

    @Autowired
    @Qualifier("userServiceDatabaseImpl")
    private UserService userService;

    public boolean authenticate(String username
        return userService.authenticate(use
    }
}
```

Como vemos la anotación @Qualifier nos permite especificar el nombre del bean que se va a inyectar en el atributo.

Para estar al pendiente sobre nuestro contenido nuevo síguenos en nuestras redes sociales <https://www.facebook.com/devs4j/> (<https://www.facebook.com/devs4j/>) y <https://twitter.com/devs4j> (<https://twitter.com/devs4j>).

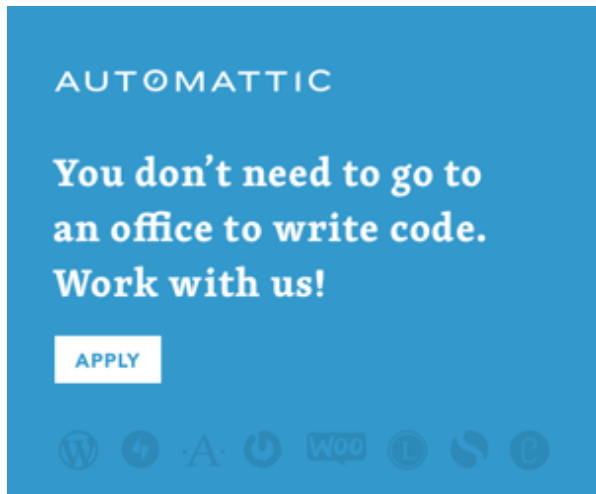
Autor: *Alejandro Agapito Bautista*

Twitter: [@raidentrance](#)

(<https://geeksjavamexico.wordpress.com/mentions/raidentrance/>)

Contacto: [raidentrance@gmail.com](mailto:raidentrance@gmail.com)

#### Anuncios



REPORT THIS AD



REPORT THIS AD

