

```
package com.example.springsecurityapplication.security;

import com.example.springsecurityapplication.models.Person;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.GrantedAuthority;
import
org.springframework.security.core.authority.SimpleGrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;

import java.util.Collection;
import java.util.Collections;

public class PersonDetails implements UserDetails {

    private final Person person;

    public PersonDetails(Person person) {
        this.person = person;
    }

    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {
        // Возвращает лист из одного элемента
        return Collections.singletonList(new
SimpleGrantedAuthority(person.getRole()));
    }

    // Позволяет получить пароль пользователя
    @Override
    public String getPassword() {
        return this.person.getPassword();
    }

    // Позволяет получить логин пользователя
    @Override
    public String getUsername() {
        return this.person.getLogin();
    }

    // Аккаунт действителен
    @Override
    public boolean isAccountNonExpired() {
        return true;
    }

    // Аккаунт не заблокирован
    @Override
    public boolean isAccountNonLocked() {
        return true;
    }

    // Пароль является действительным/валидным
    @Override
    public boolean isCredentialsNonExpired() {
        return true;
    }

    // Аккаунт активен
    @Override
    public boolean isEnabled() {
        return true;
    }
}
```

```
}

// Метод по получанию объект пользователя
public Person getPerson(){
    return this.person;
}

}
```