User Profile Service with Caching

//UserProfileServiceApplication.java

```
package com.example.userProfile;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
@SpringBootApplication
public class UserProfileServiceApplication {
    public static void main(String[] args) {
        SpringApplication.run(UserProfileServiceApplication.class, args);
    }
}
```

//UserProfile.java

```
package com.example.userProfile.entity;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
@Entity
public class UserProfile {
    @Id
    private Long id;
    private String username;
    private String fullName;
    private String bio;
    public UserProfile() {}
    public UserProfile(Long id, String username, String fullName, String bio) {
        this.id = id;
```

```
    this.username = username;
    this.fullName = fullName;
    this.bio = bio;
}
// Getters and Setters
public Long getId() {
    return id;
}
public void setId(Long id) {
    this.id = id;
}
public String getUsername() {
    return username;
}
public void setUsername(String username) {
    this.username = username;
}
public String getFullName() {
    return fullName;
}
public void setFullName(String fullName) {
    this.fullName = fullName;
}
public String getBio() {
    return bio;
}
```

```
    public void setBio(String bio) {
        this.bio = bio;
    }
}

//UserProfileRepository.java
package com.example.userProfile.repository;
import java.util.Optional;
import org.springframework.data.jpa.repository.JpaRepository;
import com.example.userProfile.entity.UserProfile;
public interface UserProfileRepository extends JpaRepository<UserProfile, Long>
{
    Optional<UserProfile> findByUsername(String username);
}
//UserCache.java
package com.example.userProfile.cache;
import java.util.HashMap;
import java.util.Map;
import org.springframework.stereotype.Component;
import com.example.userProfile.entity.UserProfile;
@Component
public class UserCache {
    private Map<String, UserProfile> cache = new HashMap<>();
    public void save(UserProfile profile) {
        cache.put(profile.getUsername(), profile);
    }
    public UserProfile get(String username) {
```

```
        return cache.get(username);
    }
}

//UserProfileService.java
package com.example.userProfile.service;

import com.example.userProfile.cache.UserCache;
import com.example.userProfile.entity.UserProfile;
import com.example.userProfile.repository.UserProfileRepository;
import io.github.resilience4j.circuitbreaker.annotation.CircuitBreaker;
import jakarta.annotation.PostConstruct;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class UserProfileService {
    @Autowired
    private UserProfileRepository repository;
    @Autowired
    private UserCache cache;
    @CircuitBreaker(name = "userProfileCB", fallbackMethod =
"getUserProfileFromCache")
    public UserProfile getUserProfile(String username) {
        simulateDatabaseIssue(username);
        UserProfile profile = repository.findByUsername(username)
                .orElseThrow(() -> new RuntimeException("User not found"));
```

```
        cache.save(profile);
        return profile;
    }
    @PostConstruct
    public void init() {
        repository.save(new UserProfile(1L, "mina", "Minakshi Patel", "Java
developer from Pune"));
    }
    public UserProfile getUserProfileFromCache(String username, Throwable ex) {
        UserProfile cached = cache.get(username);
        if (cached != null) return cached;
        UserProfile fallback = new UserProfile();
        fallback.setUsername(username);
        fallback.setFullName("Unavailable");
        fallback.setBio("Data temporarily unavailable");
        return fallback;
    }
    private void simulateDatabaseIssue(String username) {
        if (username.equalsIgnoreCase("error")) {
            throw new RuntimeException("Simulated DB failure");
        }
    }
}
```

//UserProfileController.java

```
package com.example.userProfile.controller;
```

```
import com.example.userProfile.entity.UserProfile;
import com.example.userProfile.service.UserProfileService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
@RestController
@RequestMapping("/api/users")
public class UserProfileController {
    @Autowired
    private UserProfileService service;
    @GetMapping("/{username}")
    public UserProfile getUserProfile(@PathVariable String username) {
        return service.getUserProfile(username);
    }
}
```