

Praca domowa 08 – control

Szlachetka Miłosz
Nr albumu. 114014

1. Cel zadania

Celem zadania była implementacja komponentu EJB realizującego funkcjonalność licznika. Ponadto należało zaimplementować servlet, który przyjmuje żądania i na ich podstawie steruje procesem obliczeń (zliczania).

2. Struktura programu

IControlRemote - Interfejs typu remote, który prezentuje się następująco:

```
@Remote
public interface IControlRemote {
    public void start();
    public void stop();
    public void increment(int i);
    public int counter();
    public int errors();
}
```

ControlImpl - komponent EJB typu stateful implementujący interfejs IControlRemote. Realizuje funkcjonalność licznika. Uzasadnieniem tego, że komponent został opatrzony adnotacją @Stateful jest to, że musi on przetrzymywać pewien stan, który jest związany z poszczególnymi użytkownikami. Komponent ten posiada pole "private boolean active" przechowujące informację o jego stanie (możliwe 2 stany - zliczanie(active = true) oraz wstrzymanie zliczania(active = false)). Dodatkowo zawiera zmienne "private int counter", "private int errorCounter" służące do przetrzymywania stanu licznika poprawnych jak i niepoprawnych żądań.

Metody:

-ControlImpl() - konstruktor, ustawiający wartości liczników na 0 oraz ustalania stanu na wstrzymanie (active=false)

-void start() - ustawia komponent w stan zliczania. W przypadku gdy komponent znajduje się już w stanie zliczania, zwiększa licznik błędów o 1.

-void stop() - ustawia komponent w stan wstrzymania. W przypadku gdy komponent znajduje się już w stanie wstrzymania, zwiększa licznik błędów o 1.

-void increment(int i) - jeśli komponent jest w stanie zliczania inkrementuje licznik (counter) o wartość i. Jeśli komponent jest w stanie wstrzymania, zwiększa licznik błędów o 1.

-int counter() - zwraca aktualną wartość licznika (counter)

-int errors() - zwraca aktualną wartość licznika błędów (errorCounter)

Control - servlet odpowiedzialny za odbieranie i przetwarzanie żądań klienta. Steruje procesem zliczania. Do zliczania wykorzystuje komponent EJB **ControllImpl**.

Metody:

-IControlRemote connectToEJB() - za pomocą interfejsu JNDI i mechanizmu lookup wyszukuje komponent EJB **ControllImpl** i zestawia z nim połączenie.

-void accessEJB(HttpSession session) - metoda pobierająca komponent EJB z sesji użytkownika, jeśli komponent EJB się w niej znajduje. Jeśli komponent nie jest zapisany w sesji, to jest zestawiane połączenie z nowym komponentem przy użyciu metody "connectToEJB", a następnie otrzymany EJB jest zapisywany do sesji użytkownika, w celu późniejszego pobrania.

-void handleStateReq(String stateVal) - odpowiada za przetwarzanie żądania zawierającego parametr "state". Jako argument przyjmuje wartość parametru "state"

-String register(HttpServletRequest request) - rejestruje i przetwarza żądanie. W zależności od parametrów URL wykonuje określone akcje zliczania przy użyciu komponentu "ControllImpl". Jako parametr przyjmuje żądanie HTTP. Zwraca odpowiedź dla klienta.

-void processRequest(HttpServletRequest request, HttpServletResponse response) - odbiera żądanie od klienta, pobiera komponent EJB dla danego klienta (metoda **accessEJB**), przetwarza żądanie (metoda **register**) i zwraca wynik.

3. Mechanizm wyszukiwania i zestawiania połączenia

Wyszukiwanie komponentu EJB zostało zrealizowane dzięki interfejsowi JNDI, który daje możliwość wyszukiwania i uzyskiwania dostępu do serwisów i katalogów przy użyciu przenośnych nazw. Aby uzyskać dostęp do komponentu EJB o nazwie **ControllImpl** i użyto następującej nazwy, która określa położenie komponentu:

java:global/114014/ControllImpl!IControlRemote

java:global - przestrzeń nazw, do wyszukiwania zdalnych EJB.

ControllImpl - nazwa komponentu EJB

!IControlRemote - interfejs komponentu

Połączenie z komponentem EJB było zrealizowane poprzez utworzenie kontekstu (InitialContext), a następnie wywołanie na nim metody lookup, z parametrem będącym adresem komponentu.

Dla każdego żądania klienta (request) w servlecie sprawdzana była zawartość sesji. Jeśli w sesji klienta nie znajdował się komponent EJB **ControllImpl** to nawiązywano z nim połączenie w sposób opisany powyżej, a następnie komponent był zapisywany do sesji. Jeśli natomiast w sesji użytkownika (HttpSession) znajdowała się komponent EJB, to zostawał on pobrany. Dzięki takiemu mechanizmowi, każdy klient jest związany ze swoim własnym komponentem EJB **ControllImpl**.

4. Przetwarzanie żądań

Servlet rejestruje i przetwarza żądania, które rozpoznawane są poprzez parametry zawarte w URL. Parametrami żądań mogą być:

- login - obecność tego parametru w URL powoduje wywołanie metody start na komponencie ControlImpl
- logout - obecność tego parametru w URL powoduje wywołanie metody stop na komponencie ControlImpl
- state - obecność tego parametru w URL powoduje wywołanie metody increment na komponencie ControlImpl. Metoda increment jest wywoływana z argumentem będącym wartością parametru state, jeśli takową wartość posiada. W innym przypadku argumentem metody increment jest wartość 1.
- result - obecność tego parametru w URL powoduje pobranie wartości licznika (counter) oraz licznika błędów (errorCounter) z komponentu EJB oraz obliczenie różnicy między otrzymanymi wartościami (`controlEJB.counter() - controlEJB.errors()`).