Podstawy aplikacji sieciowych

Zadanie: Aplikacja Web, kontener i komponenty na przykładzie platformy Jakarta EE

Autorzy (2x imię, nazwisko, numer indeksu):

*UWAGA: Zadanie należy zrealizować prowadząc interakcję z aplikacją za pomocą klienta Httpie (dostępny jako pakiet httpie w dystrybucjach Linux; polecenie http). Dla uwidocznienia wszystkich informacji używaj opcji –v.*

# Część 1: Wdrażanie aplikacji i jej kontekst

Dokonaj wdrożenia aplikacji (załączony plik WAR) na serwerze aplikacyjnym Payara. Zachowaj parametry domyślne. Poniżej wstaw zrzut ekranu z opisem wdrożonej aplikacji (w tym widocznym kontekstem aplikacji)

Miejsce na zrzut ekranu (usuń ten tekst po jego wstawieniu)

Zmodyfikuj archiwum aplikacji dodając/modyfikując odpowiedni deskryptor wdrożenia tak, aby kontekst aplikacji uległ zmianie (sugerowane skrócenie). Zaprezentuj odpowiedni fragment deskryptora.

Miejsce na listing (fragment deskryptora). Zachowaj oryginalne formatowanie tej sekcji.

Dokonaj wdrożenia tak zmodyfikowanej aplikacji na serwerze aplikacyjnym Payara. Zachowaj parametry domyślne. Wykaż dostępność aplikacji w nowym kontekście żądaniem do jej kontekstu bez wskazania zasobu/komponentu.

Miejsce na listing (realizacja żądania). Zachowaj oryginalne formatowanie tej sekcji.

# Część 2: Wymiana danych tekstowych

Aplikacja zawiera serwlet *ParamServlet* o kontekście */Param*. Serwlet ten rozpoznaje parametr *name* prezentując jego wartość.

Wykonaj trzy żądania do tego serwletu: nie przekazując parametru *name*, przekazując go żądaniem typu GET oraz żądaniem typu POST.

Miejsce na listing (realizacja żądania bez parametru). Zachowaj oryginalne formatowanie tej sekcji.

Miejsce na listing (realizacja żądania z parametrem, metoda GET). Zachowaj oryginalne formatowanie tej sekcji.

Miejsce na listing (realizacja żądania z parametrem, metoda POST). Zachowaj oryginalne formatowanie tej sekcji.

Zmodyfikuj archiwum aplikacji dodając stronę (plik HTML) zawierającą formularz zbudowany tak, aby jego przesłanie spowodowało wygenerowanie żądania typu POST do serwletu z przekazaniem wprowadzonej wartości parametru. Zaprezentuj kod HTML strony.

Miejsce na listing (nowa strona HTML). Zachowaj oryginalne formatowanie tej sekcji.

Dokonaj wdrożenia tak zmodyfikowanej aplikacji. Posługując się przeglądarką zademonstruj poprawne działanie strony wstawiając dwa zrzuty ekranu: z widoczną stroną i wypełnionym formularzem oraz z widoczną stroną wygenerowaną przez serwlet.

Miejsce na zrzut ekranu (formularz) (usuń ten tekst po jego wstawieniu)

Miejsce na zrzut ekranu (wynik) (usuń ten tekst po jego wstawieniu)

# Część 3: Wymiana danych binarnych

Aplikacja zawiera serwlet *FileUploadServlet* o kontekście */FUS*. Serwlet ten rozpoznaje parametr tekstowy *algo* który jest interpretowany jako oznaczenie algorytmu skrótu zgodnie ze specyfikacją klasy MessageDigest. Ponadto serwlet ten rozpoznaje parametr plikowy *file* który jest interpretowany jako zawartość przesyłanego pliku, dla którego obliczana jest i prezentowana suma kontrolna zgodnie z zawartością parametru *algo*.

Wykonaj trzy żądania do tego serwletu: jedno z poprawnymi wartościami obu parametrów (wskaż dowolny plik dostępny w systemie), jedno z pominiętym parametrem *algo* oraz jedno z nieprawidłową wartością parametru *algo*. Zaobserwuj, w jaki sposób serwlet sygnalizuje klientowi błąd wykonania.

Miejsce na listing (realizacja żądania z poprawnymi parametrami). Zachowaj oryginalne formatowanie tej sekcji.

Miejsce na listing (realizacja żądania z pominiętym parametrem). Zachowaj oryginalne formatowanie tej sekcji.

Miejsce na listing (realizacja żądania z nieprawidłowym parametrem). Zachowaj oryginalne formatowanie tej sekcji.

Aplikacja zawiera serwlet *FileDownloadServlet* o kontekście */FDS*. Serwlet ten wysyła klientowi zawartość binarną określonego typu. Wykonaj żądanie do tego serwletu i na podstawie wyniku określ typ wysyłanej zawartości.

Miejsce na listing (realizacja żądania). Zachowaj oryginalne formatowanie tej sekcji.

Zmodyfikuj archiwum aplikacji dodając stronę (plik HTML) zawierającą wyświetlenie danych generowanych przez serwlet zgodnie z ich formatem. Zaprezentuj kod HTML strony.

Miejsce na listing (nowa strona HTML). Zachowaj oryginalne formatowanie tej sekcji.

Dokonaj wdrożenia tak zmodyfikowanej aplikacji. Posługując się przeglądarką zademonstruj poprawne działanie strony wstawiając zrzut ekranu z widoczną stroną.

Miejsce na zrzut ekranu (usuń ten tekst po jego wstawieniu)

# Część 4: Zasięgi

Aplikacja zawiera serwlet *ScopeServlet1* o kontekście */Scope1*. Wykonaj dwa żądania do tego serwletu, zaobserwuj generowane identyfikatory sesji oraz ciasteczka. Dlaczego za każdym razem generowana była nowa sesja?

Miejsce na listing (dwa kolejne żądania). Zachowaj oryginalne formatowanie tej sekcji.

Wykonaj dwa kolejne żądania, tym razem przesyłając do serwletu odpowiedniej wartości ciasteczka tak, aby odwołać się do utworzonych w poprzednim eksperymencie sesji. Zweryfikuj, że za każdym razem identyfikator sesji jest zgodny z informacją zawartą w ciasteczku.

Serwlet ten rozpoznaje parametr tekstowy *store* który jest interpretowany jako wartość, która powinna być zapamiętana przez serwlet. Wykonaj dwa kolejne żądania, odwołując się do utworzonych w poprzednim eksperymencie sesji, przesyłając dwie różne wartości tego parametru.

Miejsce na listing (dwa kolejne żądania z przesłaniem parametru). Zachowaj oryginalne formatowanie tej sekcji.

Następnie wykonaj dwa kolejne żądania, odwołując się do utworzonych w poprzednim eksperymencie sesji, bez przesyłania parametru, zweryfikuj zapamiętanie wartości parametru odrębnie dla obu sesji.

Miejsce na listing (dwa kolejne żądania z weryfikacją zapamiętanej wartości). Zachowaj oryginalne formatowanie tej sekcji.

Powtórz dwa poprzednie kroki (ustawienie parametru dla obu sesji, weryfikacja wartości parametru dla obu sesji – NIE ZMIENIAJ KOLEJNOŚCI) dla serwletów o kontekstach */Scope2* i */Scope3*. Na podstawie wyników określ, z jakich zasięgów składowania wartości korzystają te serwlety.

Miejsce na listing (dwa kolejne żądania z przesłaniem parametru oraz dwa kolejne żądania z weryfikacją zapamiętanej wartości, dla serwletu /Scope2). Zachowaj oryginalne formatowanie tej sekcji.

Miejsce na Twoją odpowiedź (z jakiego zasięgu korzysta ten serwlet?). Zachowaj oryginalne formatowanie tej sekcji.

Miejsce na listing (dwa kolejne żądania z przesłaniem parametru oraz dwa kolejne żądania z weryfikacją zapamiętanej wartości, dla serwletu /Scope3). Zachowaj oryginalne formatowanie tej sekcji.

Miejsce na Twoją odpowiedź (z jakiego zasięgu korzysta ten serwlet?). Zachowaj oryginalne formatowanie tej sekcji.

Ponownie odwołaj się do jednej z utworzonych w poprzednim eksperymencie sesji wykonując żądanie do serwletu */Scope1* ustawiając dowolną wartość parametru *quit*.

Miejsce na listing (żądanie z przesłaniem parametru quit dla serwletu /Scope1). Zachowaj oryginalne formatowanie tej sekcji.

Następnie wykonaj dwa kolejne żądania do serwletu */Scope1*, odwołując się do utworzonych w poprzednim eksperymencie sesji, bez przesyłania parametru, zweryfikuj zapamiętanie wartości parametru odrębnie dla obu sesji. Jaka operacja została wykonana w reakcji na przesłanie parametru *quit*?

Miejsce na listing (dwa kolejne żądania z weryfikacją zapamiętanej wartości), dla serwletu /Scope1. Zachowaj oryginalne formatowanie tej sekcji.

Miejsce na Twoją odpowiedź (jaka operacja została wykonana w jednej z sesji). Zachowaj oryginalne formatowanie tej sekcji.