

Desarrollo de Aplicaciones Web

Tema IX: Sesiones

Temas de la Unidad

- Estado de la Sesión del Cliente
- Estado de la Sesión en el Servidor
- Estado de la Sesión en la Base de Datos

Estado de la Sesión del Cliente

Estado de la Sesión del Cliente

- Algunas aplicaciones permiten
 - Colocar toda la información de la sesión del cliente
 - Para luego enviarla al servidor
 - Con esto se puede utilizar un Servidor sin Estado
- El cliente también necesita almacenar datos
 - Si el app es de cliente enriquecido
 - Se puede hacer dentro de su propia estructura
 - Con los campos de su interfaz

Estado de la Sesión del Cliente

- Los objetos no visuales utilizan una estrategia mejor
 - Estrategias tales como DTO (Data Transfer Objects)
 - Modelo de Dominio
- De cualquier manera no representa un problema
- Si se trabaja con una interfaz HTML
 - Sube la dificultad pero es manejable

Estado de la Sesión del Cliente Implementación

- Existe 3 formas para manejarlo
 - Parámetros URL
 - Campos Ocultos
 - Cookies

Prámetros URL

- Son fáciles si la cantidad de datos es baja
- El estado de la sesión está representado por
 - Una serie de parámetros en cada consulta
 - Puede utilizar algún tipo de ID
 - Muchos ambientes re-escriben el URL
 - Para agregar el identificador correspondiente
 - El cambio constante del URL puede ocasionar
 - Problemas para guardar la dirección de la página
 - Cuando el usuario quiere crear un bookmark del sitio

Campos Ocultos

- Es un campo enviado al buscador
 - Este no se despliega en la página Web
 - Se usa el valor "hidden" en la propiedad "type"
 - Correspondiente al elemento input
- La idea es ocultar los datos de la sesión
 - Todo en el campos ocultos
 - En los cuales se agrega lo que sea necesario
 - Es necesario aplicar algún tipo de formato
 - Tal como XML
 - Formato de texto
- Es más versátil sin importar la cantidad de info

Cookies

- Son enviadas en ambos sentidos al servidor
 - Se mueven en ambas direcciones automáticamente
- Cada cookie tiene un tamaño determinado
 - Esto limita la cantidad de información por cada una
 - Algunos usuarios no les gusta su uso y las desactiva
 - No son más seguras que otras soluciones
 - Trabajan únicamente en un mismo dominio
- Si se deshabilitan las cookies en el sistema
 - Algunas plataformas pueden cambiar de esquema
 - Moviéndose a "Parámetros URL"

Estado de la Sesión del Cliente

Ventajas y Desventajas

- Es un mecanismo ideal para servidores sin estado
 - Esto maximiza la cantidad de concurrencia
 - Si el cliente falla toda la información se pierde
 - Pero el usuario esperaría que así fuera
- Un problema es manejar muchos datos
 - Al transferir muchos datos aumenta el retraso
 - Disminuye el tiempo de respuesta del app
 - Si la cantidad de datos es poca, el esquema funciona

Estado de la Sesión del Cliente

Ventajas y Desventajas

- La seguridad representa un problema
 - Los datos son vulnerables a ser
 - Vistos
 - Alterados
 - Se puede corregir con encriptación
 - Pero esto aumenta la complejidad
 - Aumenta el tiempo de espera
 - Se maneja normalmente un identificador de sesión
 - Que se encuentra del lado del cliente
 - Lo bueno es que no es vulnerable pues es un número aleatorio

Estado de la Sesión en el Servidor

Estados de la Sesión del Servidor

- Aquí los datos de la sesión están en el servidor
 - Almacenado en la memoria del servidor
 - Estos datos están asociados a un identificador
 - El identificador correspondiente a cada sesión
 - Cada cliente debe proporcionar su ID de sesión
 - Envía su ID de sesión por cada solicitud
 - El servidor localiza los datos, los recupera y entrega

Estados de la Sesión del Servidor Implementación

- Se debe crear un objeto en la RAM del servidor
 - Este maneja los datos de la sesión
 - Se puede tener una especie de mapa (Map)
 - Con los objetos de sesión ingresados por un ID
 - El cliente provee solamente el ID
 - El objeto de la sesión puede recuperar los datos
 - Todo a partir del mapa para procesar la solicitud

Estados de la Sesión del Servidor Implementación

- Este método es intensivo en la RAM del servidor
 - El servidor debe tener suficientes recursos
 - Para manejar las tareas simultáneas
 - Supone además que el servidor atiende una sola tarea
 - El servidor no debe trabajar el cluster
 - Si el servidor falla
 - La sesión se debe abandonar
 - Todo el trabajo realizado se pierde

Estados de la Sesión del Servidor Implementación

- Este ambiente puede ser problemático
 - Depende de la aplicación y sus necesidades
 - Se puede arreglar esto para cada app
 - Pero puede agregar demasiada complejidad
 - Se supone que el ambiente es simple
 - Tanto para su desarrollo
 - Como para su mantenimiento
 - El primer problema trabajar con la RAM del servidor
 - La memoria está en poder de los objetos de sesión
 - Esta representa la principal objeción para usar esta solución
 - Lo ideal es serializar el estado de la sesión
 - En un almacenamiento persistente
 - Esto libera la memoria y puede manejar más concurrencia

Estados de la Sesión del Servidor Implementación

- El serializar debe ser simple
 - La fortaleza de este método es su simplicidad
- Varias plataformas ofrecen un mecanismo
 - Para serialización binaria
 - Ejecuta la serialización del grafo del objeto
 - Todo mediante un método fácil
- Se puede serializar utilizando otros métodos
 - Utilizando texto plano
 - Utilizando lenguajes conocidos como archivos XML

Estados de la Sesión del Servidor Implementación

- La serialización binaria puede generar
 - Problemas de legibilidad a simple vista
 - Puede haber problemas con el control de versión
 - Si se modifica una clase en un campo
 - Después que se haya serializado
 - Puede ser que no se pueda leer de vuelta el objeto

Estados de la Sesión del Servidor Implementación

- Otro aspecto es donde almacenar
 - La sesión del servidor
 - Se puede hacer en una base de datos local del servidor
 - En el sistema de archivos del servidor de aplicaciones
 - Lo anterior lo hace fácil pero
 - No permite la utilización de clusters
 - O el failover de forma eficiente
 -

Estados de la Sesión del Servidor Implementación

- Otro aspecto es donde almacenar
 - Se puede almacenar en un lugar de acceso general
 - Localizado en un servidor compartido
 - Habilitando la posibilidad del uso de clusters
 - Y la respuesta automatizada a fallos
 - Esto aumenta
 - El costo de la solución
 - El manejo
 - La complejidad del ambiente en general

Estados de la Sesión del Servidor

Ventajas y Desventajas

- Los datos de la sesión se pueden almacenar
 - Durante largos períodos si es necesario
 - No es tan importante cuando eliminarlos
 - Si es necesario contar con un mecanismo
 - Para manejar las sesiones
 - Revisando las sesiones activas
 - Eliminando las que hayan expirado
- Cuando todo se maneja en el servidor
 - El programador no debe enviar tanta información
 - El paso de información con el cliente es mínima
 - El esfuerzo solo aumenta cuando hay clusters

Estado de la Sesión en la Base de Datos

Estado de la Sesión en la Base de Datos

- La última opción es almacenar la sesión
 - Dentro de una base de datos del servidor
 - Esto mantendrá la información de cada sesión
 - Debidamente almacenada en un registro de la BD

Estado de la Sesión en la Base de Datos: Implementación

- Las llamadas van del cliente hacia el servidor
 - El objeto servidor extrae los datos necesarios
 - Con ellos realiza la solicitud a la BD
 - Se ejecuta le trabajo solicitado
 - Los datos se guardan de nuevo en la BD
 - Se muestra los resultados al cliente

Estado de la Sesión en la Base de Datos: Implementación

- En el proceso el objeto servidor extrae
 - Información de la sesión del cliente
 - Número de ID de la sesión
 - El conjunto apropiado de claves
 - Para extraer los datos apropiados de la base de datos
 - Los datos asociados son una combinación de datos
 - Datos de la sesión que son datos locales a la interacción actual
 - Datos confirmados, relevantes para la interacción completa

Estado de la Sesión en la Base de Datos: Implementación

- Los datos de la sesión son locales
 - No deben afectar otras partes del sistema
 - Al menos hasta confirmar las modificaciones
 - Si se desea guardar el estado temporal del objeto
 - Este no debe afectar el objeto final
 - El objeto final se almacena en la BD
 - El estado temporal no debe aparecer en la BD
 - En la BD se puede añadir un campo
 - Este maneja la sesión en ejecución
 - Este campo puede ser booleano "Pendiente"

Estado de la Sesión en la Base de Datos: Implementación

- Otra opción es usar el campo ID “Pendiente”
 - Esto hace fácil el encontrar todo lo relacionado a la sesión
- Las consultas que requieren solo registros
 - Pueden tener una cláusula para la sesión
 - ID IS NULL
 - O se puede utilizar una vista para filtrar los datos

Estado de la Sesión en la Base de Datos: Implementación

- El uso del campo ID de sesión es invasivo
 - Toda transacción con la BD debe conocer el campo
 - Así puede saber que utilizar sin confundir datos
 - Las vistas pueden evitar el problema de invasión
 - Pero tienen un alto costo de tiempo
- Otra opción es separar la tablas
 - Manteniendo aparte la tabla de datos pendientes
 - Las tablas de pendientes pueden
 - Evadir o reforzar las reglas según sea necesario

Estado de la Sesión en la Base de Datos: Implementación

- Al guardar datos pendientes
 - No se aplican reglas de validación
 - Con base en el período de sesiones
 - Ya que esto se maneja en la lógica del objeto
- Las tablas pendientes son clones de las reales
 - Esto hace el proceso transparente
 - Mantiene la lógica de asignación similar a la real
 - Debe tener los mismos campos que la tabla real
 - Se debe agregar el ID de sesión
 - Con esto se encuentran todos los datos de una sesión

Estado de la Sesión en la Base de Datos: Implementación

- Si una sesión se cancela o abandona
 - Debe haber un proceso de limpieza de la tabla
 - Con base en el ID de sesión
 - Encuentra los datos relacionados a la sesión
 - Borra con base a la búsqueda lo necesario
 - Si el usuario abandona la sesión sin informarlo
 - Se necesita algún mecanismo de tiempo real
 - Esto se puede solucionar con un demonio de sistema
 - El demonio busca las sesiones antiguas
 - Para ello la tabla debe tener además la hora de inicio
 - Así el demonio puede tomar decisiones al respecto

Estado de la Sesión en la Base de Datos: Ventajas y Desventajas

- La BD provee el manejo de
 - Sesiones concurrentes
 - Conexiones a clusters
 - Facilitando su manejo y correcto trabajo
 - Permite hacer uso de pooling
 - Los objetos trabajan sin estado en el servidor
- Si no hay estado de sesión y
- Se puede guardar los datos como un registro
 - Esta solución es la opción óptima para elegir

Estado de la Sesión en la Base de Datos: Ventajas y Desventajas

- Se pierde el tiempo en cada I/O con la BD
 - Por las solicitudes que entran y salen a la BD
 - Se puede reducir un poco el costo por el caché
 - Pero siempre hay castigo de tiempo por escrituras
- El esfuerzo de programación es más alto
 - Todo alrededor del manejo del estado de sesión