Введение в Java EE. Сервлеты

№ урока: 5 **Курс:** Основы Java EE

Средства обучения: InteliJ Idea, MySQL Workbench

Обзор, цель и назначение урока

Знакомство с Атрибутами и параметрами. Что такое веб контейнер, контекст веб приложения, cookie, сессия. Знакомство с JSP и JSTL.

Изучив материал данного занятия, учащийся сможет:

- Поймет разницу между атрибутами и параметрами и как их применять
- Работать с контекстом приложения. Обращаться с разных страниц к одному объекту.
- Хранить объекты в сессии во время работы приложения.
- Обрабатывать форму с помощью JSP и скриплетов
- Обрабатывать форму с помощью JSTL

Содержание урока

1. Понятия контекст, сессия, запрос.
2. Разница между атрибутами и параметрами.
3. Понятия ServletContext и ServletConfig.
4. Понятия Cookies.
5. Понятия JSP.
6. JSP, скримлеты, стандартные теги.
7. Знакомство с JSTL

Резюме

- **Веб-приложение** — клиент-серверное приложение, в котором клиент взаимодействует с сервером при помощи браузера, а за сервер отвечает — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами.
- **Сессия** – в информатике, в частности в сети, сеанс представляет собой полупостоянный интерактивный обмен информацией, также известный как диалог, беседа или встреча, между двумя или более коммуникационными устройствами или между компьютером и пользователем (см. Сеанс входа в систему). Сессия настраивается или устанавливается в определенный момент времени, а затем сбрасывается в какой-то более поздний момент. Установленный сеанс связи может включать в себя более одного сообщения в каждом направлении. Сеанс обычно, но не всегда, имеет состояние, означающее, что по крайней мере одна из сообщающихся частей должна сохранять информацию о истории сеанса, чтобы иметь возможность общаться, в отличие от передачи без гражданства, где сообщение состоит из независимых запросов с ответы.
  Установленный сеанс является основным требованием для выполнения связи, ориентированной на соединение. Сессия также является основным шагом для передачи в режимах без установления соединения. Однако любая однонаправленная передача не определяет сеанс.
  Коммуникационный транспорт может быть реализован как часть протоколов и служб на уровне приложения, на уровне сеанса или на транспортном уровне в модели OSI.

CyberBionic systematics

CyberBionic Systematics ® 2018
19Eugene Sverstyuk Str.,5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Примеры применения:
HTTP-сессии, которые позволяют связывать информацию с отдельными посетителями
Сеанс удаленного входа telnet
Пример уровня сеанса:
Интернет-звонок на основе протокола инициирования сеанса (SIP)
Пример транспортного слоя:
Сеанс TCP, который является синонимом виртуальной сети TCP, TCP-соединения или установленного TCP-сокета.
В случае транспортных протоколов, которые не реализуют формальный уровень сеанса (например, UDP) или где сеансы на прикладном уровне, как правило, очень недолговечны (например, HTTP), сеансы поддерживаются программой более высокого уровня с использованием метода, определенного в обмене данными. Например, обмен HTTP между браузером и удаленным хостом может включать в себя HTTP-файл cookie, который идентифицирует состояние, например, уникальный идентификатор сеанса, информацию о предпочтениях пользователя или уровне авторизации.
Предполагалось, что HTTP / 1.0 допускает только один запрос и ответ во время одного сеанса Web / HTTP. Версия протокола HTTP / 1.1 улучшила это, завершив интерфейс Common Gateway (CGI), упростив обслуживание веб-сеанса и поддержку HTTP-файлов cookie и загрузки файлов.
Большинство сеансов клиент-сервер поддерживается транспортным уровнем - одно соединение для одного сеанса. Однако каждая фаза транзакции сеанса Web / HTTP создает отдельное соединение. Для поддержания непрерывности сеанса между этапами требуется идентификатор сеанса. Идентификатор сеанса встроен в ссылки <A HREF> или <FORM> динамических веб-страниц, чтобы он возвращался обратно в CGI. Затем CGI использует идентификатор сеанса для обеспечения непрерывности сеанса между этапами транзакции. Одним из преимуществ одного соединения в фазе является то, что он хорошо работает по низкоскоростным (модемным) соединениям.

- **Ку́ки** (англ. *cookie*, буквально — печенье) — небольшой фрагмент данных, отправленный веб-сервером и хранимый на компьютере пользователя. Веб-клиент (обычно веб-браузер) всякий раз при попытке открыть страницу соответствующего сайта пересылает этот фрагмент данных веб-серверу в составе HTTP-запроса. Применяется для сохранения данных на стороне пользователя, на практике обычно используется для:
  аутентификации пользователя;
  хранения персональных предпочтений и настроек пользователя;
  отслеживания состояния сеанса доступа пользователя;
  ведения статистики о пользователях.
  Приём браузерами куки требуют многие сайты с ограничениями доступа, большинство интернет-магазинов. Настройка оформления и поведения многих веб-сайтов по индивидуальным предпочтениям пользователя тоже основана на куки.
  Куки легко перехватить и подменить (например, для получения доступа к учетной записи), если пользователь использует нешифрованное соединение с сервером. В группе риска пользователи, выходящие в интернет при помощи публичных точек доступа Wi-Fi и не использующие при этом таких механизмов, как SSL. Шифрование позволяет также решить и другие проблемы, связанные с безопасностью передаваемых данных.
  Имеется и ряд заблуждений о куки. Они главным образом основаны на уверенности людей, что куки являются компьютерными программами. На самом деле, куки — это простые текстовые данные, набор символов, передаваемый при запросах к веб-сайту, и они не могут выполнять какие-либо действия самостоятельно. В частности, куки не могут быть ни вирусами, ни шпионскими программами. Таким образом, куки могут быть опасны только в плане деанонимизации и слежения за действиями пользователя.
  Большинство современных браузеров позволяют пользователям выбрать — принимать куки или нет, но их отключение делает невозможной работу с некоторыми сайтами.
- Интерфейс **javax.servlet.ServletConfig** используется для передачи конфигурационной информации сервлету. Каждый сервлет имеет свой собственный ServletConfig, за создание которого отвечает контейнер сервлетов.
- Для доступа из сервлета к параметрам WEB-приложения необходимо использовать интерфейс javax.servlet.ServletContext. Объект ServletContext является уникальным и доступен всем сервлетам.

CyberBionic systematics
CyberBionic Systematics ® 2018
19Eugene Sverstyuk Str.,5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

ServletContext позволяет получить доступ к параметрам WEB-приложения, определенным в дескрипторе web.xml тегом <context-param>.
Объект **ServletContext** можно получить с помощью метода getServletContext() интерфейса ServletConfig
Интерфейс ServletContext определяет доступ к следующим функциям для работы с аттрибутами:
public Object getAttribute(String name)
public java.util.Enumeration getAttributeNames()
public void setAttribute(String name, Object object)
public void removeAttribute(String name)
Роль аттрибутов может выполнять объект любого класса. Цель данных функций связана с пересылкой между несвязанными друг с другом сервлетами разных объектов.

- Отличия **ServletConfig** от **ServletContext**:
  Объект ServletConfig является уникальный для каждого сервлета, а ServletContext определен для всего приложения;
  ServletConfig используется для получения параметров инициализации сервлета, а ServletContext для получения параметров инициализации приложения для всех сервлетов;
  Объект ServletConfig не позволяет устанавливать параметры/атрибуты, в то время как их можно установить в объекте ServletContext, которые становятся доступными всем сервлетам.
  То есть, можно сказать, что ServletConfig индивидуален для каждого сервлета, а ServletContext - для WEB-приложения и доступен всем сервлетам.
- **JSP** (**JavaServer Pages**) — технология, позволяющая веб-разработчикам создавать содержимое, которое имеет как статические, так и динамические компоненты. Страница JSP содержит текст двух типов: статические исходные данные, которые могут быть оформлены в одном из текстовых форматов HTML, SVG, WML, или XML, и JSP- элементы, которые конструируют динамическое содержимое. Кроме этого могут использоваться библиотеки JSP-тегов, а также EL (Expression Language), для внедрения Java-кода в статичное содержимое JSP-страниц.
  Код JSP-страницы транслируется в Java-код сервлета с помощью компилятора JSP-страниц Jasper, и затем компилируется в байт-код виртуальной машины java (JVM). Контейнеры сервлетов, способные исполнять JSP-страницы, написаны на платформонезависимом языке Java. JSP-страницы загружаются на сервере и управляются из структуры специального Java server packet, который называется Java EEWeb Application. Обычно страницы упакованы в файловые архивы .war и .ear.
  JSP является платформонезависимой, переносимой и легко расширяемой технологией для разработки веб-приложений.
- Стандартная библиотека тегов JSP (англ. ***JavaServer Pages Standard Tag Library***, **JSTL**) — расширение спецификации JSP, добавляющее библиотеку JSP тегов для общих нужд, таких как разбор XMLданных, условная обработка, создание циклов и поддержка интернационализации. JSTL — конечный результат JSR 52, разработанного в рамках процесса сообщества Java.
  8 мая 2006 был выпущен релиз JSTL 1.2.
  JSTL является альтернативой такому виду встроенной в JSP логики, как скриплеты, то есть прямые вставки Java кода. Использование стандартизованного множества тегов предпочтительнее, поскольку получаемый код легче поддерживать и проще отделять бизнес-логику от логики отображения.

Закрепление материала

- Что такое сессия?
- Что такое cookie?
- В чем разница между сессией и cookie?
- В чем разница между классами ServletConfig и ServletContext?
- В чем разница между атрибутами и параметрами?
- Что такое JSP?
- Что такое скриплеты?
- Что такое JSTL?

CyberBionic systematics
CyberBionic Systematics ® 2018
19Eugene Sverstyuk Str.,5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Дополнительное задание

Задание
Сделать форму регистрации клиента. При успешной регистрации на странице отображать всех зарегистрированных пользователей (имя, возраст, телефон).

Самостоятельная деятельность учащегося

Задание 1
Выучите основные понятия, рассмотренные на уроке.

Задание 2
Добавить к форме из дополнительного задание CSS и данные добавлять в БД carsshop.

Задание 3
Из домашнего задания из первого урока Calculator добавить возможность отображать операции всех клиентов. В левой части отображается список клиента. В правой список всех клиентов, сгруппированный по session id. Порядок обращений клиентов должен сохранятся.

Рекомендуемые ресурсы

JSP
https://uk.wikipedia.org/wiki/JSP

JSP Tutorial
https://www.tutorialspoint.com/jsp/index.htm

JSTL
https://www.tutorialspoint.com/jsp/jsp_standard_tag_library.htm

Работа с Cookie
http://crypto.pp.ua/2010/06/cookie-v-java/

CyberBionic systematics

CyberBionic Systematics ® 2018
19Eugene Sverstyuk Str.,5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com