**1. Методы сервлета. Жизненный цикл сервлета.**

Интерфейс сервлета содержит 5 методов:

- init(ServletConfig config) - вызывается только 1 раз при загрузке сервлета контейнером сервлетов, используется для инициализации данных сервлета;

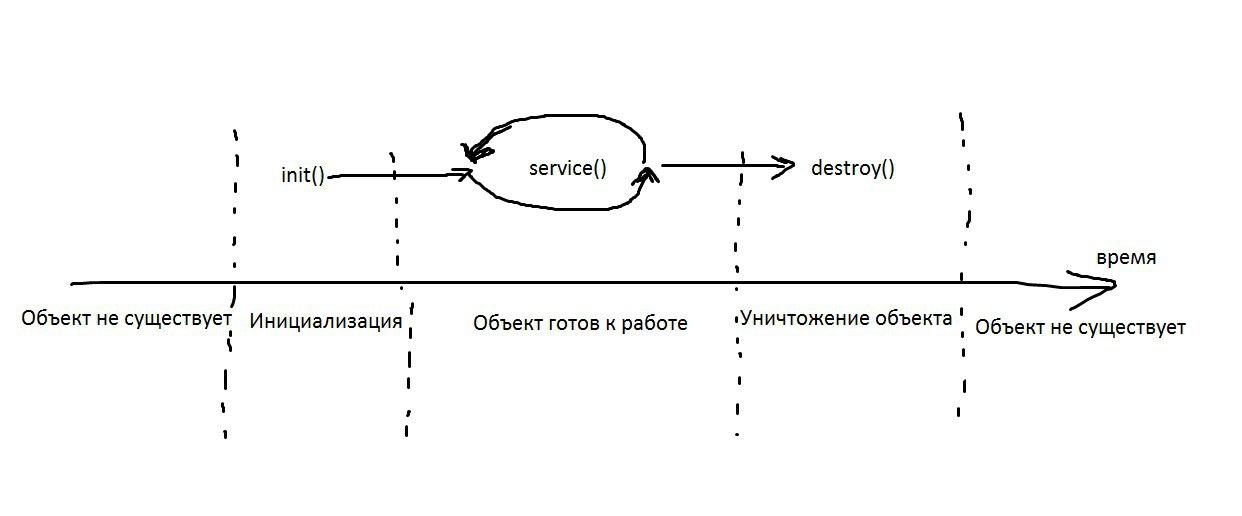
- service(ServletRequest req, ServletResponse resp) - основной метод сервлета, он вызывается при каждом запросе. В качестве аргументов принимает входящий запрос (ServletRequest) и исходящий ответ сервлета клиенту (ServletResponse). Именно в данном методе осуществляется формирование ответа;

- destroy() - вызывается только 1 раз при уничтожении сервлета контейнером сервлетов, используется для инициализации данных сервлета;;

- getServletConfig() - возвращает объект, содержащий информацию о конфигурации сервлета;

- getServletInfo() - возвращает строку с информацией о сервлете.

Жизненным циклом сервета управляет контейнер сервлетов, цикл можно схематично представить:

Изначально при создании объекта сервлета вызывается один раз метод init(), после этого объект готов к использованию и вызывается каждый раз при получении запроса от клиента, перед уничтожением объекта сервлета вызывается метод destroy() в котором можно произвести закрытие ресурсов.

**2. Что такое jsp. Жизненный цикл.**

JSP – это технология, позволяющая динамически генерировать HTML страницы. Жизненный цикл JSP страницы состоит из следующих этапов:

- трансляция jsp страницы;

- компиляция jsp страницы;

- загрузка скомпилированного класса;

- создание экземпляра класса;

- вызов метода jspInit на созданном экземпляре класса один раз для инициализации jsp. Данный метод можно перегрузить, используя декларации;

- вызов метода \_jspService для каждого запроса на созданном экземпляре класса;

- вызов метода jspDestroy на созданном экземпляре класса один раз перед уничтожением объекта контейнером сервлета.

**3. Чем отличается методы POST и GET. Если не указать напрямую, какой из этих методов выполнится по умолчанию?**

Можно выделить несколько различий между методами POST и GET:

- при использовании метода POST параметры передаются в теле HTTP запроса, а при использовании метода GET – параметры передаются как часть URL;

- в методе GET ограничен объем передаваемой информации (параметров);

- с помощью метода POST можно передавать файлы, в отличие от метода GET;

- при использовании метода GET существует риск того, что поисковый робот может выполнить тот или иной открытый запрос.

**4. В чём различие forward и redirect?**

Forward отличается от redirect тем, что:

- forward выполняется на стороне сервера клиент (браузер) об этом ничего не знает, а redirect выполняется на стороне клиента, т.к. клиент получает ответ со статусом 302 и отправляет еще один запрос по указанному URL;

- при forward отправляется 1 запрос, при redirect – 2;

- при forward исходный запрос доступен в конечном ресурсе, а при redirect нет, т.к. создается новый запрос;

- при forward можно использовать только ресурсы данного сервера, при redirect можно не только внутри сервера;

- при forward клиент получает данные быстрее, т.к. осуществляется только 1 вызов меотда, при redirect дольше, т.к. в итоге информацию получает 2умя запросами;

- при forward клиент не видит ресурс на который был выполнен переход, при redirect новый адрес видно в адресной строке браузера.

**5. Из чего состоит url?**

URL состоит из:

- протокола передачи данных (наиболее распространенные: https://, http://, ftp://)

- префикс www (может отсутствовать).

- субдомен, может отсутствовать, обычно используется, если необходимо на одном сайте совместить несколько принципиально разных тематик;

- домен (чаще всего название компании, бренда);

- доменная зона (например, ru, com, org и т.д);

- порт (чаще всего не указывается, в таком случае используется значение по умолчанию “80”);

- путь к ресурсу

- параметры (набор пар “ключ=значение”), начинается со знака “?”, а пары разделяются знаком “&”.

**6. Что нужно написать в строке браузера, чтобы обратиться к хосту, на котором установлен tomcat, развёрнуто приложение, в котором есть несколько сервлетов? Как обратиться к конкретному сервлету? Что такое www? Где нужно указывать порт?**

Чтобы обратиться к хосту, необходимо указать его ip адрес или можно использовать доменное имя, затем через “:” указать порт, на котором работает tomcat, затем указать “/” и имя приложения (папки из webapps, в котором размещено нужное приложение), а затем указать “/” и путь к необходимому сервлету. Путь к сервету задается указанием маппинга сервета и URL-pattern в файле WEB-INF/web.xml.

WWW – это приставка, которую можно указывать перед доменным именем сайта, означающая world wide web (т.е. система, предоставляющая доступ к текстовой информации по протоколу http). В настоящее время, необходимости в указании данного префикса нет.

**7. Что такое HTTP? Отличия HTTP 1.0 и HTTP 2.**

HTTP – это протокол прикладного уровня для передачи информации между клиентом и сервером.

Различия HTTP 1.0 и HTTP 2 заключаются в следующем:

- HTTP 2 может отправлять множественные запросы параллельно по единому соеденению TCP, это сокращает время прохождения сигнала в обе сторны и тем самым сокращает время загрузки сайта;

- HTTP 2 сжимает ненужные хедеры;

- HTTP 2 появилась возможность передавать не текстовую информацию, а бинарную;

- в HTTP 2 появилась возможность отправки данных по инициативе сервера;

- в HTTP 2 появилась возможность приоритезации.

**8. Cохраняет ли http протокол своё состояние.**

Http протокол не сохраняет свое состояние между парами “запрос-ответ”, поэтому каждому запросу приходится включать в себя всю необходимую серверу информацию, без учёта прошлых запросов. Но при этом компоненты, использующие HTTP могут самостоятельно сохранять информацию.

**9. Отличие авторизации от аутентификации.**

Аутентификация – это процесс подтверждения личности зарегистрированного пользователя, т.е. другими словами это проверка учетных данных (напирмер, логина и пароля). Если идентификатор и пароль совпадают с записями, хранящимися в базе данных системы, пользователю предоставляется доступ. В случае неправильного ввода данных программа вызывает предупреждение безопасности и блокирует вход.

Авторизация – это процесс проверки доступа аутентифицированного пользователя к определенному ресурсу, это необходимо когда в приложении есть ресурсы к которым может иметь доступ только пользователь с определенной ролью (например, к разделу блокировки пользователей должен иметь доступ толкьо пользователь с ролью “Администратор”).

**10. Сервер приложений. Веб-сервер. Отличия.**

Веб-сервер – это сервер, принимающий HTTP запросы и выдающий им HTTP ответы (обычно с HTML страницей, файлом, изображением и др.).

Сервер приложений – сервер, который преднозначени для функционирования приложений, он обеспечивает их жизненный цикл, может работать как “веб-сервер”. Выполняет такие функции как создание программной среды, идентификацию и авторизацию клиентов, создание для каждого клиента сессии и т.д.

Отличие заключается в том, что "веб-сервер" может обрабатывать запросы только на статическую информацию, и он будет передавать запросы на динамический контент на "сервер приложений", а также “веб-сервер” работает только с протоколом HTTP, а сервер приложений поддерживает и другие протоколы.

**11. Пошагово рассказать, что происходит, когда пользователь нажимает на кнопку. (с формы поля сетаются в request, потом вызывается контейнер сервлетов, потом он как-то по request понимает, куда нужно идти дальше (в дескриптор развертывания, а их может быть несколько, нужно как-то понимать в какой).**

Когда пользователь нажимает на кнопку, происходит отправка запроса браузером по указанному адресу, в этом запросе передаются параметры, введенные пользователем. Данный запрос получает контейнер сервлетов, он создает новый поток для обработки данного запроса, а также 2 объекта: HttpServletRequest (содержащий информацию поступившего запроса) и HttpServletResponse (преднозначенный для формирования обратного ответа клиенту). Далее контейнер определяет с помощью дескриптора web.xml (в котором указывается маппинг обработчиков и их адрес) к какому обработчику пришел запрос (если данный обработчик ранее не вызывался, то происходит его загрузка, создание объекта, инициализация, а для jsp еще и компиляция, и трансляция) и вызывает метод service() (или для jsp \_jspService()), который в своей стандартной реализации вызовет в зависимости от метода запроса метод doGet(), doPost() или др.. В данном методе происходит обработка полученного запроса и формирование ответа. После этого контейнер сервлетов из объекта HttpServletResponse формирует Http ответ, который будет отправлен клиенту.