Оглавление

[HTTP 2](#_Toc25692912)

[Заголовки 2](#_Toc25692913)

[HTTP метод 2](#_Toc25692914)

[HTTP Response Status 3](#_Toc25692915)

[HTTP Response Content type (тип содержимого) 3](#_Toc25692916)

[Что делает сервер? 3](#_Toc25692917)

[Сервлет 3](#_Toc25692918)

[Контейнер сервлетов 4](#_Toc25692919)

[Жизненный цикл севрлета 4](#_Toc25692920)

[URL‑параметры 4](#_Toc25692921)

[Redirect 4](#_Toc25692922)

[Forward 5](#_Toc25692923)

[Разница Redirect и Forward 5](#_Toc25692924)

[Session 5](#_Toc25692925)

[Cookies 6](#_Toc25692926)

[Как работают сессии? 7](#_Toc25692927)

HTTP

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5. | Клиент  Сервер (URL)  request (запрос, HTTP‑request)  response (ответ, HTTP‑response) |

У HTTP‑запросов есть некоторая структура: Headers (заголовок) и тело

Заголовки

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15. | General Headers («Основные заголовки»)  Request URL: https://www.youtube.com/  Request Method: GET  Status Code: 200  Request Headers («Заголовки запроса»)  Response Headers («Заголовки ответа»)  content type: text/html; charaset=utf-8  status: 200  date: Web, 07 Mar 2018 13:33:42 GMT  location: https://www.youtube.com/  set-Cookie: <cookie-name>=<cookie-value>  Entity Headers («Заголовки сущности») |

HTTP метод

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22. | GET (получаем данные)  POST (отправляем данные)  PUT  PATCH  DELETE  POST может передавать запросу файлы  Разный URL (POST безопаснее)  http://example.com?product=1234&  utm\_source=google  http://example.com  GET ограничивает объем передаваемой информации  GET поместить в закладки (есть параметры)  В GET вместо формы использовать URL  Поисковый робот может выполнить запрос GET |

HTTP Response Status

|  |  |
| --- | --- |
| 1.  2.  3.  4. | 200 все хорошо  3xx редирект  4xx ошибка клиента  5xx ошибка сервера |

HTTP Response  
Content type (тип содержимого)

|  |  |
| --- | --- |
| 1.  2.  3.  4. | text/html  text/css  text/xml  application/json |

Что делает сервер?

|  |  |
| --- | --- |
| 1.  2.  3.  3.  4.  5.  6.  7. | Сервер получает запрос и генерирует ответ для пользователя. Ответ может быть статическим или динамическим.  Клиент <‑> Сервер  Сервер <‑> Java код  Java код <‑> БД  Java код <‑> ??? |

Сервлет

|  |  |
| --- | --- |
| 1.  2.  3. | Сервлет — программа, работающая на сервере, которая взаимодействует с клиентами посредством принципа запрос-ответ |

Контейнер сервлетов

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6. | Контейнер сервлетов — программа, представляющая собой сервер, который занимается системной поддержкой сервлетов и обеспечивает их жизненный цикл в соответствии с правилами (определёнными в спецификациях) (Apache Tomcat, Jetty) |

Жизненный цикл севрлета

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10. | Жизненный цикл севрлета:  — запуск (контейнер сервлетов загружает класс сервлета, создает объект класса сервлета и вызывает метод init() сервлета)  — обслуживание запросов (контейнер сервлетов получает запрос, определяет какому сервлету запрос предназначен, передает запрос в метод service(); для каждого запроса свой поток)  — прекращение работы сервлета (контейнер сервлетов вызывает метод destroy()) |

URL‑параметры

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6. | В URL: http://example.com?product=1234&  utm\_source=google  request.getParameter("")  response.setParameter("", …); |

Redirect

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13. | Клиент ‑(запрос)‑> Сервер  Сервер ‑(ответ со статусом 302 или 301; указывается заголовок Location — куда перенаправить)‑> Клиент  Клиент –(запрос2)-> Другой URL  Другой URL ‑> Сервер  Другой URL ‑> Любой другой сервер  response.sendRedirect("URL") |

Forward

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11. | Клиент ‑(запрос)‑> Сервер  Сервер <‑> Другой URL (внутренний адрес)  Сервер –(ответ; обычно либо 200, либо ошибка)‑> Клиент  RequestDispatcher requestDispatcher =  servletContext.getRequestDispatcher(  "URL");  requestDispatcher.forward(request, response) |

Разница Redirect и Forward

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11. | Redirect происходит на клиенте (браузер совершает новый запрос), Forward происходит на сервере (клиент не знает про него)  При Redirect URL в браузере меняется, при Forward URL в браузере не меняется (перенаправление происходит целиком на сервере, клиент об этом не знает)  Redirect медленнее Forward (необходимо совершать больше операций) |

Session

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15. | Для каждого клиента должна храниться информация на сервере. Причем эта информация должна быть доступна во всех сервлетах,  30 мин  Объект на сервере, хранящий информацию о клиенте  Для идентификации сессии конкретного клиента используют Cookies.  request.getSession()  .getAttribute("")  .setAttribute("", ) |

Cookies

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25.  26.  27.  28. | Пара строк ключ-значение, которая хранится в браузере для конкретного сайта  Присылается браузером при каждом запросе к серверу  Позволяет хранить некоторые данные в памяти у клиента. Позволяет идентифицировать пользователя  Сервер добавляет новые cookies в браузер клиента или меняет существующие (или может удалить)  Когда клиент делает запрос к серверу, сервер видит пришедшие cookies  HTTP заголовок (заголовки запроса): set-Cookie: <cookie-name>=<cookie-value>  Состоит из:  Название, значение  Срок действия (поле expires) — когда удаляется из браузера  Domian, Path — к какому домену (сайту) относится. При запросе к какому адресу должна отсылаться (Same Origin Policy) |

Как работают cookie?

|  |  |
| --- | --- |
| 1.  2.  3.  4.  5.  6.  7.  8.  9.  10.  11.  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25.  26. | Клиент получает от сервера cookie с названием «JSESSIONID». Значение этого cookie — уникальный идентификатор пользователя  По этому cookie сервер узнает пользователя и узнает его сессию  Если очистить cookie в браузере, сервер больше не сможет узнать пользователя и его сессию  Cookie cookie1 = new Cookie("имя", "знач");  Cookie cookie2 = new Cookie(("имя", "знач");  cookie1.setMaxAge(24\*60\*60); // обяза-  cookie2.setMaxAge(24\*60\*60); // тельные  response.addCookie(cookie1);  response.addCookie(cookie2);  Cookie[] cookies = request.getCookies();  cookies[i].getName();  cookies[i].getValue();  cookie2.setMaxAge(0); // удаление  // удаление после закрытия браузера  cookie2.setMaxAge(-1); |