



DataArt

Java-интенсив



Chapter 7.1

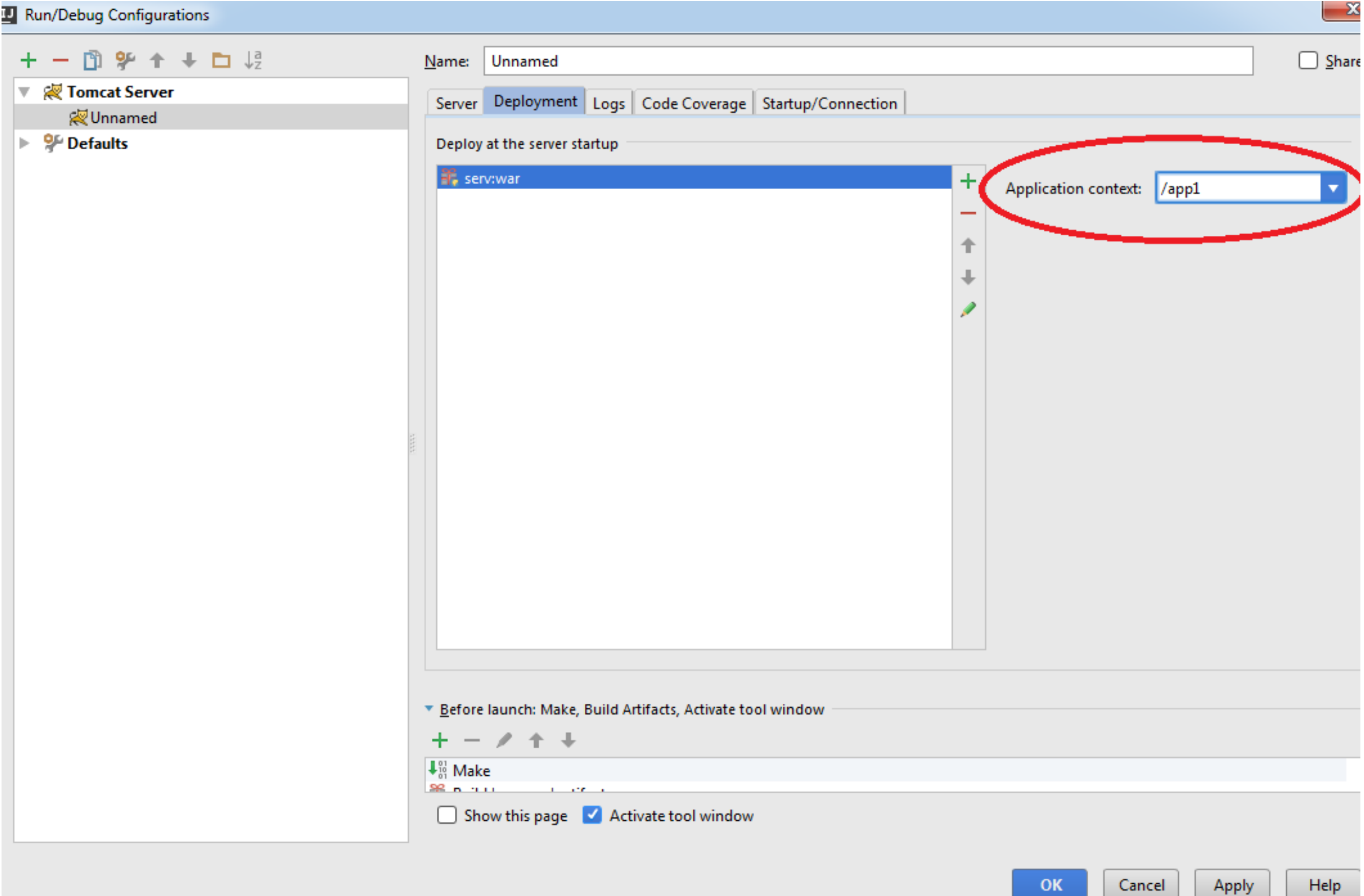
Servlets and Tomcat.



Context Path

- В целом вы можете задеплоить в томкат более одного приложения
- Многие из вас столкнулись с тем, что их приложения по умолчанию деплоились не в корневую папку (и в целом это хороший подход)
- В момент деплоя можно посмотреть, какую структуру имеет внутренняя папка `deploy` в томкате

Context Path



Context Path



Локальный диск (D:) ▶ apache-tomcat-8.5.6 ▶ webapps ▶

Справка

Добавить в библиотеку ▾ Общий доступ ▾ Записать на оптический диск Новая папка

Имя	Дата изменения	Тип	Размер
app1	26.05.2019 17:23	Папка с файлами	
docs	12.05.2018 23:09	Папка с файлами	
examples	12.05.2018 23:10	Папка с файлами	
host-manager	12.05.2018 23:10	Папка с файлами	
manager	12.05.2018 23:10	Папка с файлами	
ROOT	19.05.2019 23:37	Папка с файлами	
WebGoat-5.2.war	12.04.2017 23:16	Файл "WAR"	36 384 КБ

Локальный диск (D:) ▶ apache-tomcat-8.5.6 ▶ webapps ▶ app1 ▶

Справка

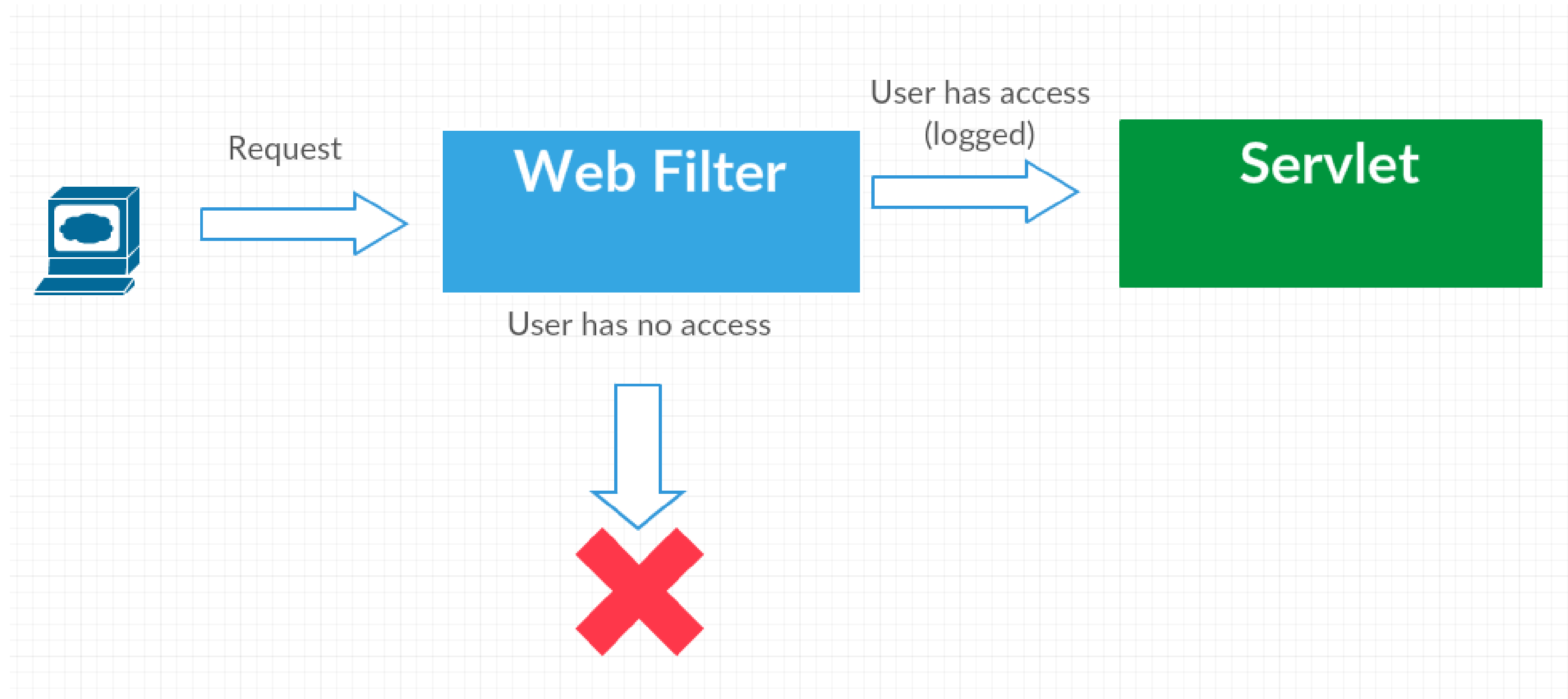
Библиотеку ▾ Общий доступ ▾ Записать на оптический диск Новая папка

Имя	Дата изменения	Тип	Размер
META-INF	26.05.2019 17:23	Папка с файлами	
WEB-INF	26.05.2019 17:23	Папка с файлами	
helloPage.jsp	19.05.2019 23:26	Файл "JSP"	1 КБ
notfound.jsp	19.05.2019 23:35	Файл "JSP"	1 КБ

Разграничение прав доступа

- Часто в приложениях требуется разграничивать права доступа
- Например, незалогиненый пользователь не может видеть ресурсы, расположенные по пути `/user/*` (то есть все, что находится в 'папке' `user`)
- Ресурсы в корневой 'папке' в нашем пример будут доступны кому угодно

Разграничение прав доступа



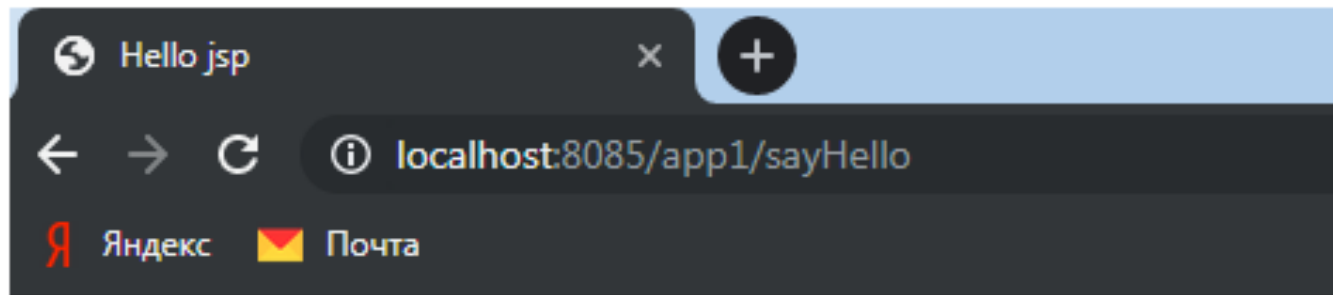
Как это реализовать?

- Как по запросу понимать, от кого он пришел?
- Где хранить информацию о том, что пользователь залогинился?

Cookies

- Чтобы узнать пользователя, используется механизм cookie
- При первом запросе на сервер, сервер выдаст вам уникальный ключ, который браузер будет прикреплять к каждому своему запросу

Cookies



hello, anonymous

Name	Headers	Preview	Response	Cookies	Timing
sayHello	<p>▼ Response Headers view source</p> <p>Content-Length: 109</p> <p>Content-Type: text/html;charset=UTF-8</p> <p>Date: Sun, 26 May 2019 14:43:06 GMT</p> <p>Set-Cookie: JSESSIONID=8E7BC84DFE637E1EA2BE0478B30E1237;path=/app1;HttpOnly</p>				
	<p>▼ Request Headers view source</p> <p>Accept: text/html,application/xhtml+xml,application/xml;q=0.</p>				

1 requests | 300 B trans...

Http Session

- Http Session механизм, с помощью которого можно запоминать данные о конкретном пользователе.
- Для каждого пользователя (а мы уже знаем один из способов определять, от кого пришли запросы)
- В нашем случае Session - это некоторое подобие коллекции Map, в которой можно хранить информацию о пользователе (например, его имя, имеет ли он доступ к каким то ресурсам)

Http Session

```
@WebServlet("/doLogin")
public class HelloWorldController extends HttpServlet {

    private LoginService loginService = new LoginService();

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String username = req.getParameter("userName");
        String password = req.getParameter("password");
        boolean success = loginService.doLogin(username, password);
        if (success) {
            req.getSession().setAttribute("userLogged", success);
            req.getRequestDispatcher("/user/success.jsp");
        } else {
            req.getRequestDispatcher("/error.jsp");
        }
    }
}
```

Http Session

```
@WebServlet("/doLogin")
public class HelloWorldController extends HttpServlet {

    private LoginService loginService = new LoginService();

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String username = req.getParameter("userName");
        String password = req.getParameter("password");
        boolean success = loginService.doLogin(username, password);
        if (success) {
            req.getSession().setAttribute("userLogged", success);
            req.getRequestDispatcher("/user/success.jsp");
        } else {
            req.getRequestDispatcher("/error.jsp");
        }
    }
}
```

Http Session

```
@WebFilter(urlPatterns = "/user/*")
public class LoginFilter implements Filter {

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {
        HttpSession httpSession = ((HttpServletRequest) request).getSession();
        Object userLogged = httpSession.getAttribute("userLogged");
        if (userLogged != null && (boolean) userLogged) {
            chain.doFilter(request, response);
        } else {
            request.getRequestDispatcher("/error.jsp").forward(request, response);
        }
    }

    @Override
    public void destroy() {
    }
}
```

Вопросы и ответы



Задание, требования

- Приложение позволяет просматривать свои ToDo – items, добавлять и удалять их, сохраняя состояние на диск (3 балла),
- В приложении реализовано разграничение прав доступа (с помощью фильтра и сессии и логин-страницы) – 2 балла
- Приложение правильно разделено на слои (View, Controller, Services, Dao) – 2 балла
- Код приложения соответствует принципам ООП, написан чисто и аккуратно (субъективный критерий) – 2 балла
- В качестве источника данных может быть использована sql база данных (в этом случае должна быть предоставлена подробная инструкция и бекап- файл) или просто файлы на диске



DataArt