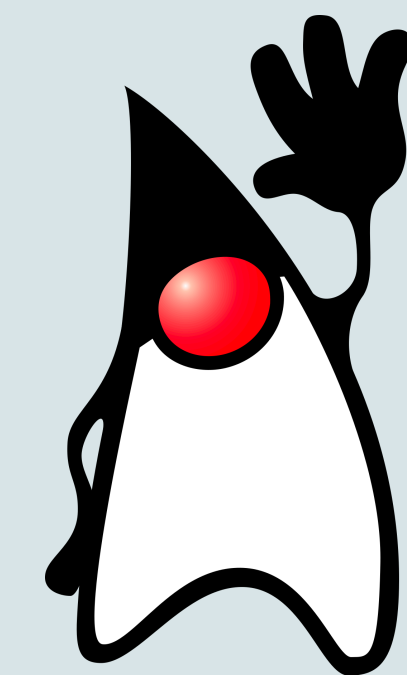




# СБЕРБАНК

---

Корпоративный  
университет



# HTML, Web и Java

Занятие №13

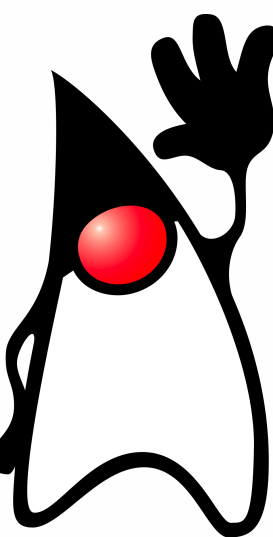


**СБЕРБАНК**

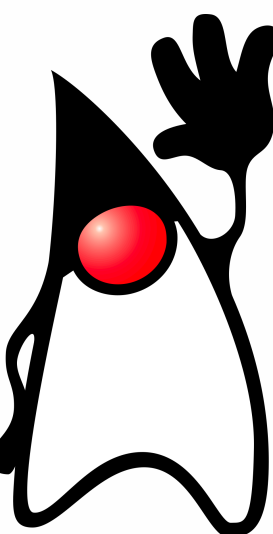
Корпоративный  
университет



- HTTP подробнее
- Servlet
- UI для библиотеки

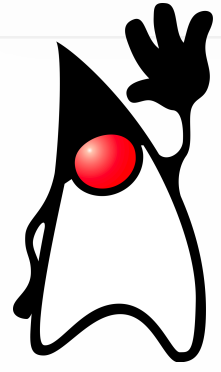


- Активно участвуем. Не стесняйтесь задавать вопрос.
- Но off-topic обсуждаем в Telegram @sb\_ku\_java\_2019\_10
- Не стесняйтесь просто спрашивать в telegram.
- ДЗ - работаем над библиотекой

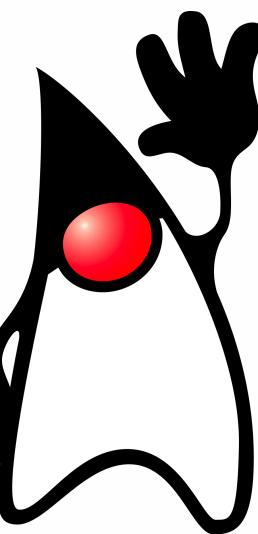




**Договорились?  
Поехали!**



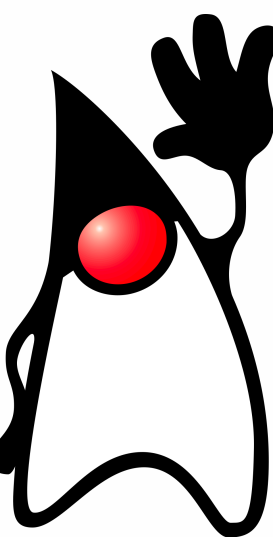
- **HTTP подробнее**
- Servlet
- UI для библиотеки



01

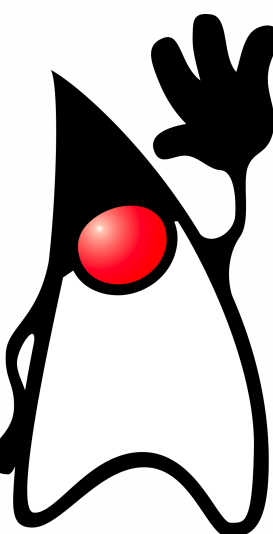
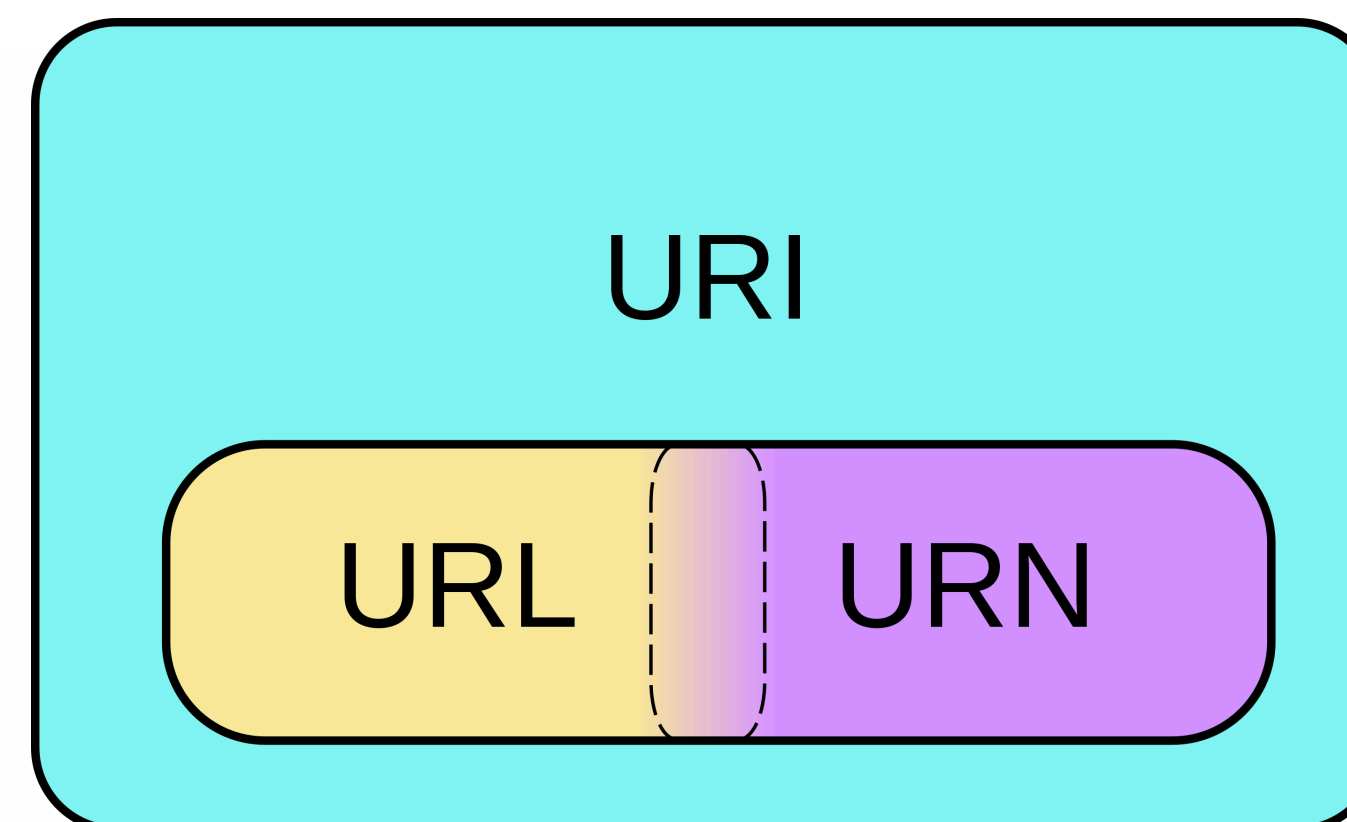
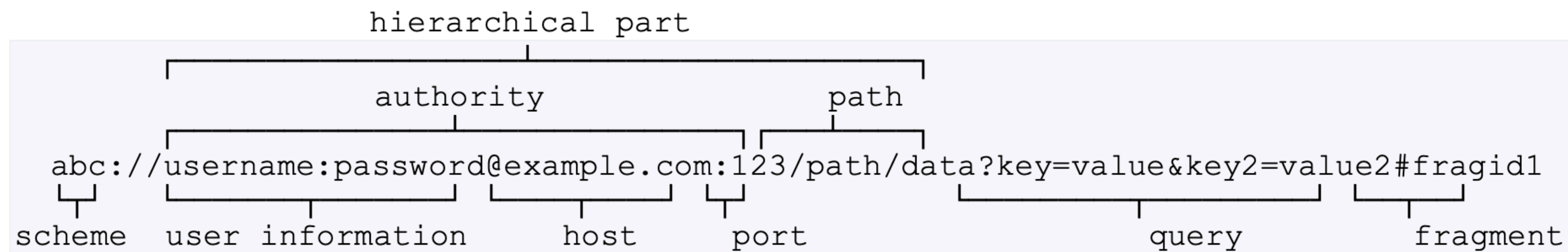
HTTP

- Исходно - статический адресуемый документ или файл в web
- Сейчас - каждый объект или сущность, которая может быть уникально идентифицирована, названа, адресована или управляема любым путем, в глобальной или локальной сети



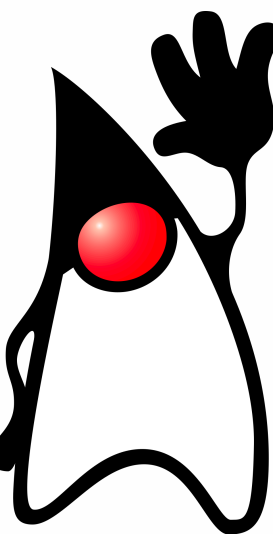


- Uniform Resource Locator
- Uniform Resource Identifier
- Uniform Resource Name



**Веб-сервер** — сервер, принимающий **HTTP**-запросы от клиентов, обычно **веб-браузеров**, и выдающий им **HTTP**-ответы, как правило, вместе с **HTML**-страницей, изображением, **файлом**, медиа-потокком или другими данными.

Веб-сервером называют как **программное обеспечение**, выполняющее функции веб-сервера, так и непосредственно **компьютер** (см.: **Сервер (аппаратное обеспечение)**), на котором это программное обеспечение работает.

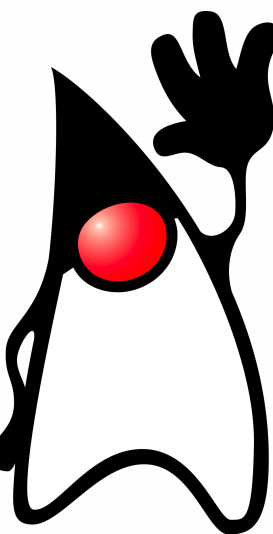


## Request message

- A request line (GET logo.png HTTP/1.1)
- Request header
- An empty line
- An optional message body

## Response message

- A status line which includes the status code and reason message (HTTP/1.1 200 OK)
- Response header
- An empty line
- An optional message body



## HTTP request header

*Accept \*/\**

*Accept-Encoding gzip, deflate*

*Accept-Language en-US,en;q=0.5*

*Cache-Control no-cache*

*Connection keep-alive*

*DNT 1*

*Host localhost:8090*

*Pragma no-cache*

*Referer http://localhost:8090/index.html*

*User-Agent Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:54.0) Gecko/20100101 Firefox/54.0*

## HTTP response header for an image

*Accept-Ranges bytes*

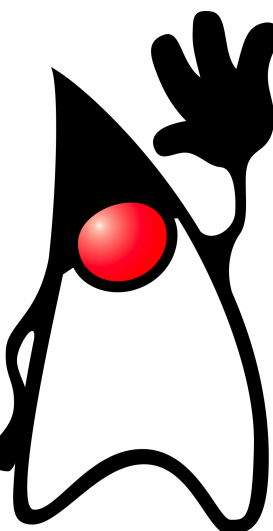
*Content-Length 13002*

*Content-Type image/png*

*Date Mon, 26 Jun 2017 13:00:10 GMT*

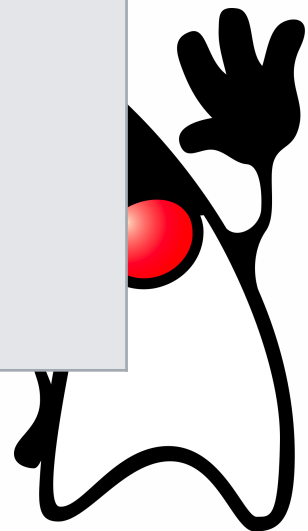
*Last-Modified Tue, 28 Feb 2017 06:13:44 GMT*

*Server Jetty(9.4.5.v20170502)*



# Коды состояния

Код	Класс	Назначение
<u>1xx</u>	<a href="#">Информационный</a> (англ. <b>informational</b> )	Информирование о процессе передачи. В HTTP/1.0 — сообщения с такими кодами должны игнорироваться. В HTTP/1.1 — клиент должен быть готов принять этот класс сообщений как обычный ответ, но ничего отправлять серверу не нужно.
<u>2xx</u>	<a href="#">Успех</a> (англ. <b>Success</b> )	Информирование о случаях успешного принятия и обработки запроса клиента. В зависимости от статуса, сервер может ещё передать заголовки и тело сообщения.
<u>3xx</u>	<a href="#">Перенаправление</a> (англ. <b>Redirection</b> )	Сообщает клиенту, что для успешного выполнения операции необходимо сделать другой запрос (как правило по другому URI). Из данного класса пять кодов <b>301</b> , <b>302</b> , <b>303</b> , <b>305</b> и <b>307</b> относятся непосредственно к перенаправлениям (редирект). Адрес, по которому клиенту следует произвести запрос, сервер указывает в заголовке <code>Location</code> . При этом допускается использование фрагментов в целевом URI.
<u>4xx</u>	<a href="#">Ошибка клиента</a> (англ. <b>Client Error</b> )	Указание ошибок со стороны клиента. При использовании всех методов, кроме <code>HEAD</code> , сервер должен вернуть в теле сообщения гипертекстовое пояснение для пользователя.
<u>5xx</u>	<a href="#">Ошибка сервера</a> (англ. <b>Server Error</b> )	Информирование о случаях неудачного выполнения операции по вине сервера. Для всех ситуаций, кроме использования метода <code>HEAD</code> , сервер должен включать в тело сообщения объяснение, которое клиент отобразит пользователю.

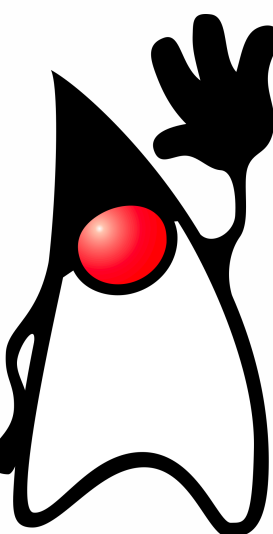


# Ваши вопросы?

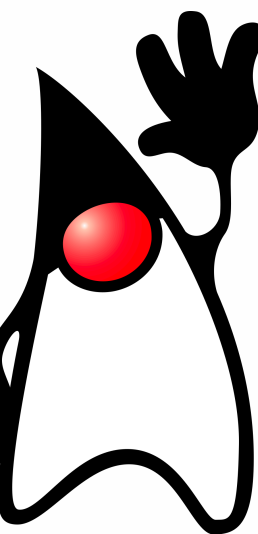
Если что — их можно задать  
ПОТОМ



- Поднимаем простейший сервер, смотрим коды состояния, смотрим куки, запросы, заголовки
- Полезные ссылки
  - <https://developer.mozilla.org/ru/docs/Web/HTTP/Overview>
  - <https://ru.wikipedia.org/wiki/HTTP>
  - <https://tools.ietf.org/html/rfc1945>



- HTTP подробнее
- **Servlet**
- UI для библиотеки



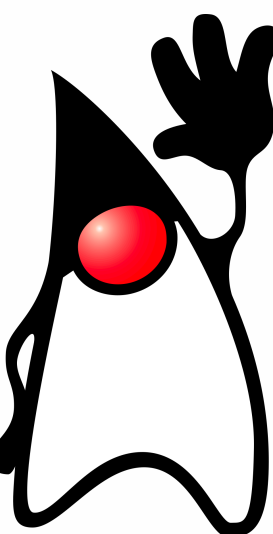
02

**Servlet**

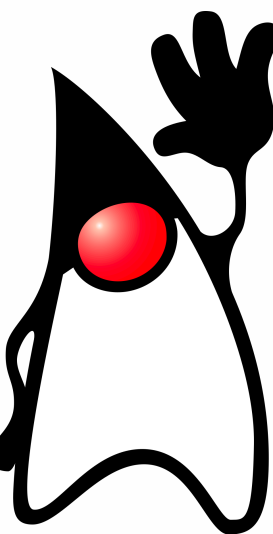
## [JSR-000369 Java Servlet 4.0 Specification Final Release](#)

A servlet is a Java™ technology-based Web component, managed by a container, that generates dynamic content.

Т.е. простыми словами, это класс, который динамически создает некий контент.



1. Клиент (например Web-браузер) отправляет запрос Web-серверу
2. Web-сервер получает запрос и передает его контейнеру сервлетов.
3. Контейнер сервлетов по запросу определяет нужный сервлет и передает ему запрос.
4. Сервлет обрабатывает запрос и формирует ответ.
5. Контейнер сервлетов передает ответ серверу, а тот - клиенту



Сервлет – это абстрактный класс HttpServlet из пакета javax.servlet.http.

Основные методы.

init

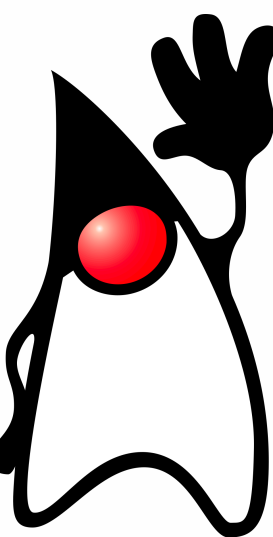
doGet

doPost

doPut

doDelete

destroy



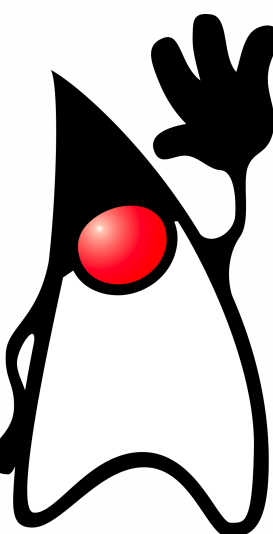


# interface HttpServletRequest

Сервлет получает запрос в виде реализации этого интерфейса.

Примеры методов:

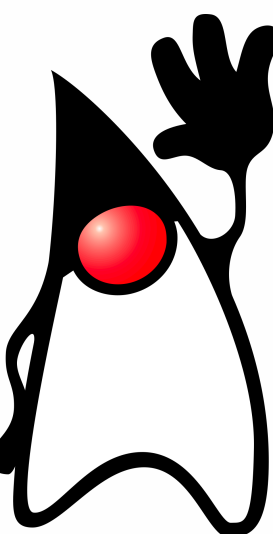
```
getCookies()  
getPathInfo();  
getUserPrincipal()  
getRequestURI();  
getSession(boolean create);  
authenticate(HttpServletRequest response)
```



ServletContext – это инфраструктурная часть, которая содержит сервлеты и прочие компоненты для обработки запросов.

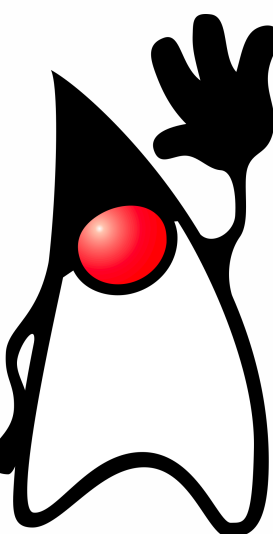
ServletConext привязывается к определенному адресу и обрабатывает все запросы, которые на этот адрес приходят.

Например, ServletContext может быть привязан к адресу <http://host/dataSource>. Контейнер сервлетов все запросы, приходящие на /dataSource будет передавать в этот ServletContext.



Filter (фильтр) – это механизм, позволяющий обработать запрос, приходящий на определенный url, и передать результат другому фильтру или сервлету.

Типичный пример применения – проверка прав доступа.



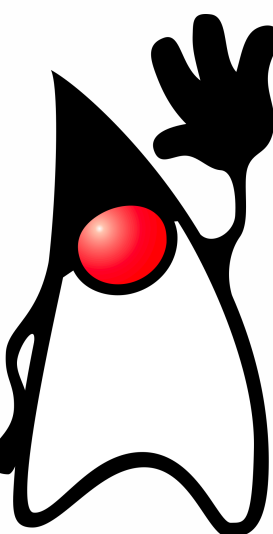
<https://www.eclipse.org/jetty/>

Eclipse Jetty provides a Web server and javax.servlet container, plus support for HTTP/2, WebSocket, OSGi, JMX, JNDI, JAAS and many other integrations.

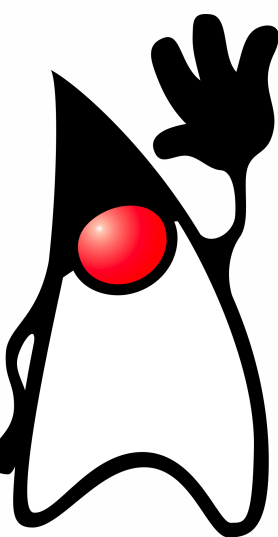
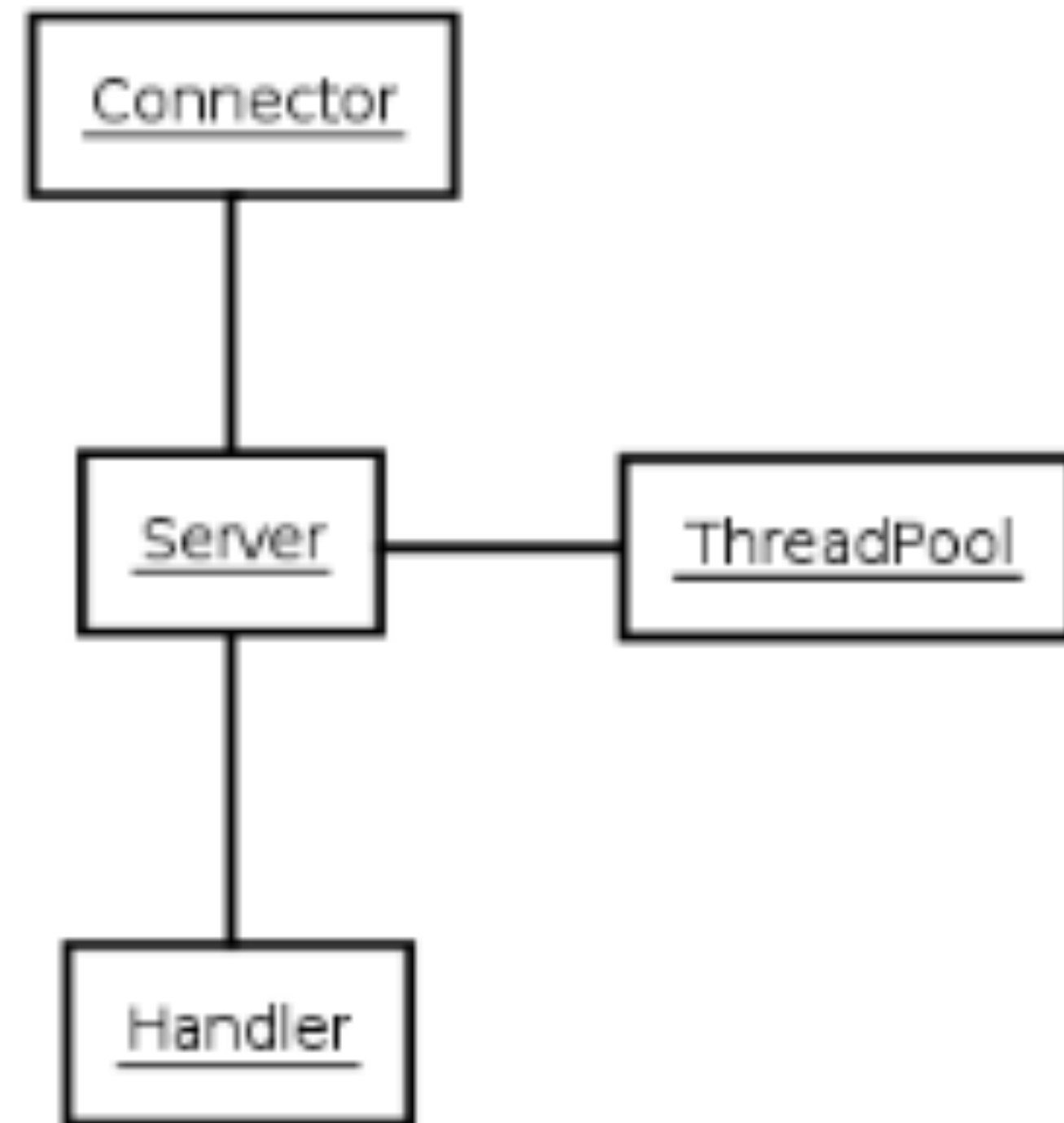
These components are open source and available for commercial use and distribution.

Это один из самых старых и популярных контейнеров сервлетов. В основном используется, как встроенный в приложение.

Контейнер сервлетов должен поддерживать спецификацию сервлетов.



## [View from 20,000 feet](#)

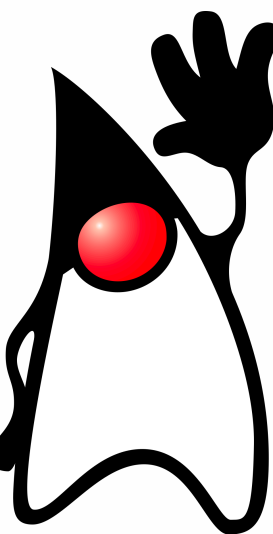


HttpServlets

Filter

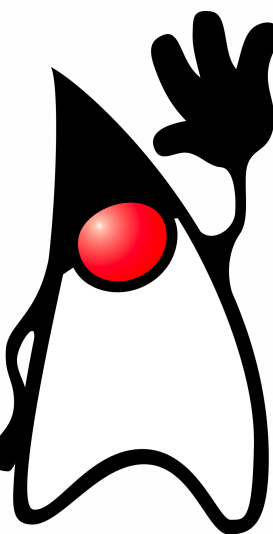
ResourceHandler (обработчик статики)

SecurityHandler (контроль доступа)

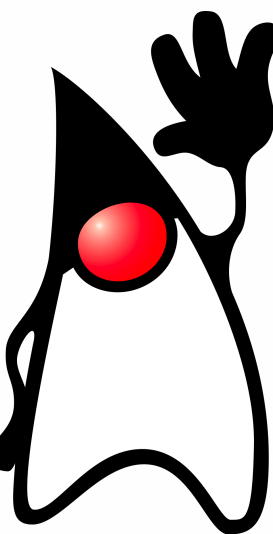




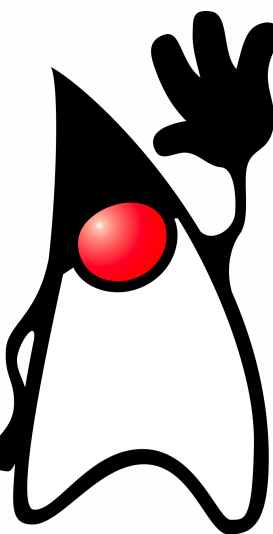
Чаще всего на Jetty делаются «нано» [RESTful-сервисы](#)



Посмотрим пример. (детали веб-сервера что видели раньше)



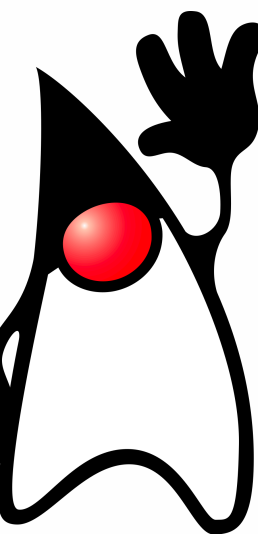
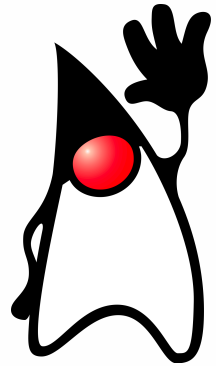
Посмотрим пример тестирования RESTful - сервиса.



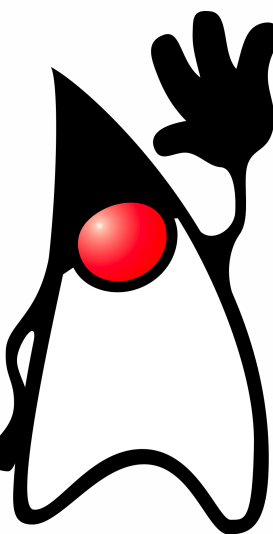
# Ваши вопросы?

Если что — их можно задать  
ПОТОМ

- HTTP подробнее
- Servlet
- **UI для библиотеки**



- Смотрим существующие ДЗ, разбираем вопросы
- Список книг на *uі*, вход с различными правами

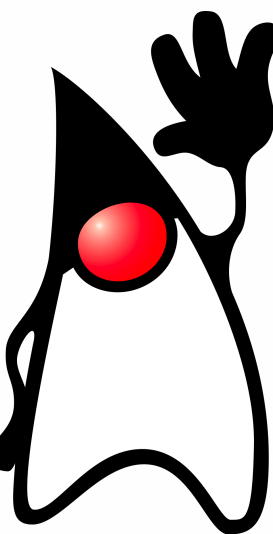




# Ваши вопросы?

Если что — их можно задать  
ПОТОМ

- Список книг
- Давайте сделаем переход книги по состоянию



**Спасибо за внимание!**

