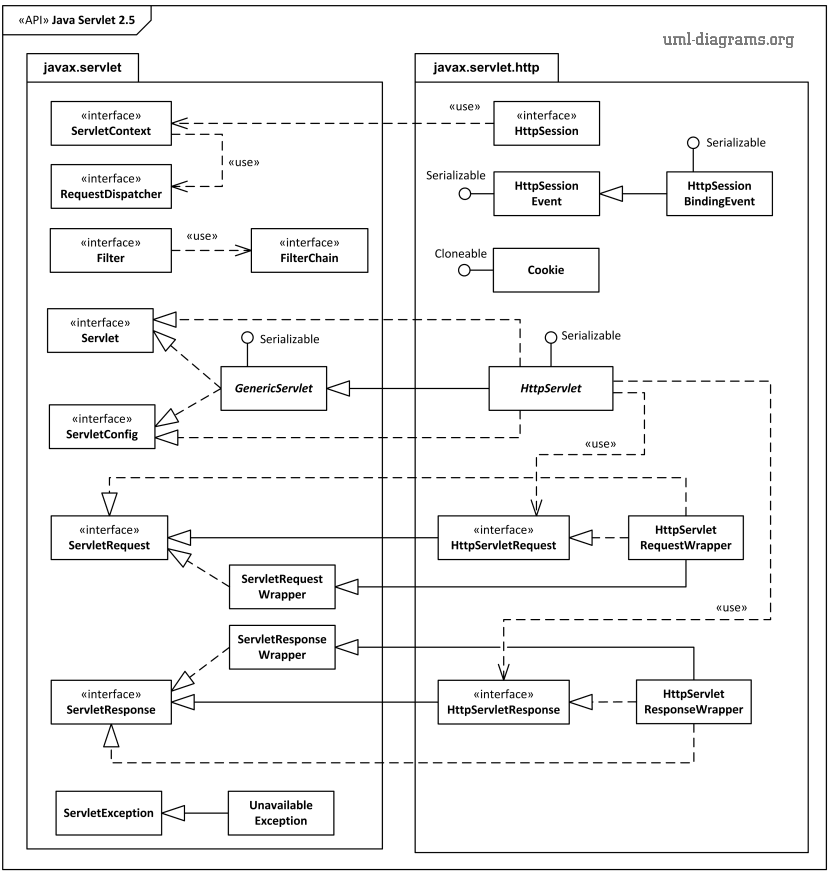
Связь между классами/интерфейсами следующая:



**GenericServlet** – тупо глушит обязательные методы **Servlet** и **ServletConfig** - что бы не захломлять код ненужными методами.

А интерфейсы **HttpServletRequest** и **HttpServletResponse** – это продвитнутые интерфейсы **ServletRequest** и **ServletResponse** только заточенные под HTTP.

Есть **ServletConfig**- который хранит мето-информацию извлечённую из web.xml (информацию созданную контейнером сервлетов) и передаёт её сервлету во время инициализации и только 1 раз, во время инициализации.

**ServletConfig** и **ServletContext**

Являются объектами создаваемыми во время инициализации сервлета (их создаёт контейнер сервлетов), оба служат для предоставления некоторых начальных параметров сервлету. Различие в том, что информация передаваемая **ServletConfig –** относится к конкретному сервлету. А информация передоваемая **ServletContext** – доступна всем сервлетам в веб-приложении.

Так прописывается имя и значение ServletContext в web.xml файле:

<context-param>

<param-name>email</param-name>

<param-value>admin@example.com</param-value>

</context-param>

Так прописывается имя и значение параметра ServletConfig для конкретного сервлета (собственно по этому в сервлете):

<servlet>

<servlet-name>Info Servlet</servlet-name>

<servlet-class>com.example.InfoServlet</servlet-class>

<!-- This is a servlet init parameter -->

<init-param>

<param-name>name</param-name>

<param-value>John Doe</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>Info Servlet</servlet-name>

<url-pattern>/test/ShowInfo.do</url-pattern>

</servlet-mapping>

Всё это можно заменить и прописывать аннотациями (смотри ссылку на статью).