Training materials

- HTTP: The Protocol Every Web Developer Must Know - Part 1.
  HTTP: протокол, который каждый разработчик должен знать (часть 1).
- Методический материал jee-eclipse-2020 (тезисное изложение).
- И. Блинов. Глава 15, стр. 456-468.
- И. Блинов. Глава 16, стр. 485-502.
- И. Блинов. Глава 18, стр. 544-558.
- YouTube канал: Java+Web в примерах на Eclipse.
- YouTube канал: Java EE для начинающих (примеры на Idea).

Code Exercise

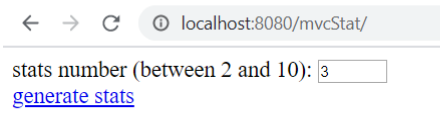
Develop a web–application using MVC pattern in accordance with the scenario below.

Scenario

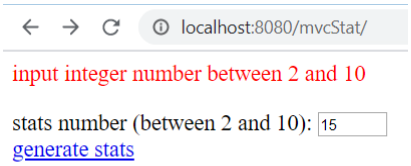
index.jsp

Controls: input box and link `generate stats`.

Main scenario. User inputs an integer value from 2 to 10 and clicks the link. Then the servlet **/start** is loaded.



Alt scenario. User inputs some wrong value (not an integer or an outside integer) and clicks the link. Then an appropriate message is appeared.



Servlet /start

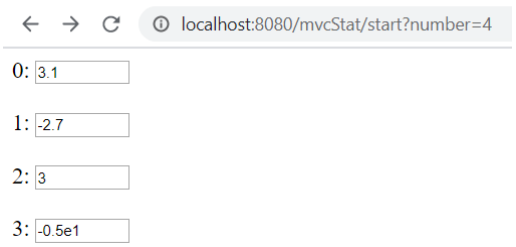
Let `number` is the request parameter meaning a statistics number.

Checks if the referrer is your app, stores `number` in the request scope and transfers the control to the page **start.jsp**.

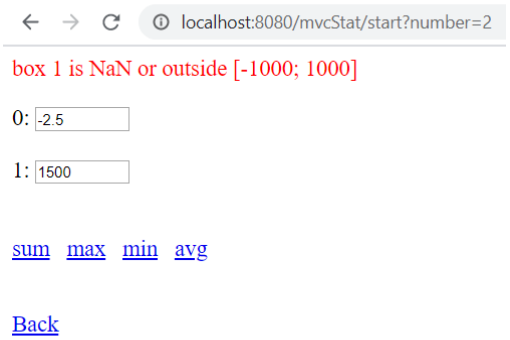
start.jsp

Controls: `number` input boxes and links `sum`, `max`, `min`, `avg`, `Back`.

Main scenario. User inputs `number` values between -1000 and +1000 and clicks the statistical operation link. Then the servlet **/result** is loaded.



Alt scenario 1. User inputs some wrong value (empty field or something with e+-. or number outside [-1000; +1000]) and clicks the statistical operation link. Then an appropriate message is appeared.



Alt scenario 2. User clicks the link `Back`. Then the page **index.jsp** is loaded.

Servlet /result

1. Checks if the referrer is your app.
2. Reads the request parameters and converts them into model data.
3. Calls the model method and gets the result.
4. Stores data in the appropriate scope.
5. Transfers the control to the page **result.jsp**.

result.jsp

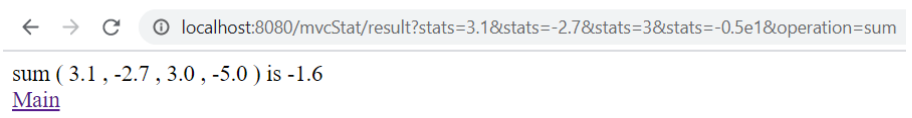
Control: link `Main`.

Main scenario. User gets the result of a statistical operation in the following format:

```
operationName ( val1 , val2 , … ) is resultValue
```

where `operationName` is `sum`, `min`, `max` or `avg`;
`val1`, `val2`, … – request values.
`resultValue` is the result of an operation.

User clicks `Main` link and returns to the **index.jsp** page.



Требования

Архив **mvcStat1-sources.zip** содержит код html-страниц и псевдокод контроллеров.

1. Исправить слой view, заменив html код на jsp код.
2. Заменить псевдокод в контроллерах на java код.
3. Создать слой model, представленный перечислением (или классом) `Operation`.

Подсказки

Слой view

На всех html-страницах имеется необходимый JavaScript код, гарантирующий отправление на сервлет валидных данных, представленных строками.

Задача заключается в замене отдельных строк тела страниц на jstl-теги, которые будут генерировать в цикле html и, возможно, проверять какие-либо условия.

Например, на странице **start.html** нужно заменить строки

```
0: <input name="stats" type="number" value="0" min="-1000" max="1000"/>
<br/><br/>

1: <input name="stats" type="number" value="0" min="-1000" max="1000"/>
<br/><br/>

2: <input name="stats" type="number" value="0" min="-1000" max="1000"/>
<br/><br/>
```

на следующий код

```
<c:forEach var="i" begin="0" end="...">
      ${i}: <input name="stats" type="number" value="0" min="-1000" max="1000"/>
      <br/><br/>
</c:forEach>
```

Очевидно, что вместо многоточия в первой строке надо указать выражение, зависящее от ввода пользователя в inputbox с `id="quantity"` на странице **index.jsp**.

На странице **result.jsp** должен появиться результат вычисления операции в заданном формате, который с точки зрения программиста, на первый взгляд, является строкой. Однако дизайнер (а jsp–страницы – это его огород) может захотеть название операции выделить мерцанием, скобки заменить на арабскую вязь, аргументы операции отображать только при наведении курсора, а результат вычисления операции должен вообще убивать наповал, что тут за дело взялся очень крутой перец :)

Вывод: атрибутами запроса являются три сущности: имя операции, коллекция или массив аргументов операции и результат вычисления операции, – а не одна строка.

На всех страницах в html-коде встречается имя проекта. Например, в index.html:

```
<form name="statsNumber" action="/mvcStat/start" onsubmit="return checkData()">
```

Jstl-тег <c:url> позволяет избавиться от этого недостатка: .

```
<form name="statsNumber" action="<c:url value='/start'/>" onsubmit="return
checkData()">
```

Слой controller

Сервлеты `StartController` и `ResultController` представлены псевдокодом.

Оба контроллера начинаются с проверки, из какого источника пришел запрос. Если источником является форма, то на сервлет придут валидные данные, представленные строками.

Если пользователь будет вводить URL сервлета, например, из адресной строки браузера, то на сервлет могут прийти невалидные данные.

Код обработки этого сценария нужно разместить в начале каждого сервлета. Он имеет вид:

```
String refferer = request.getHeader("referer");
if (refferer == null) {
      response.sendRedirect(request.getContextPath());
      return;
}
```

В данном случае запрос возвращается редиректом на страницу **index.jsp**.

Недостаток решения - копи-паст. Можно ли его исправить?

Из псевдокода сервлета `StartController` следует, что со слоем model он не взаимодействует.

В псевдокоде сервлета `ResultController` взаимодействие со слоем model реализуется созданием экземпляра перечисления `Operation` и вызовом на нем метода:

```
double result = operation.getResult(numbers);
```

Замените псевдокод в контроллерах на java код.

Слой model

Вычисление значения каждой из четырех статистических операций можно реализовать по универсальному алгоритму:

```
начальная инициализация результата;              //1
for(double value : values) {
  корректировка результата в зависимости от value;     //2
}
дополнительная обработка в случае со средним.          //3
```

Универсальный метод вычисления может быть реализован в перечислении `Operation`. В каждой константе необходимо переопределить шаг 2.

Попробуйте создать реализацию данного метода с помощью стрима, в котором шаг 2 задается лямбдой (или ссылкой на метод), переопределенной в каждой константе перечисления.