

Notes

Table of Contents

Reactor with spring	1
Linking a spring event to a Flux	1
HTMX with Spring boot and flux	1
Configure maven	1
Basic template configuration	2
Push from the backend	3
Get data at page load (lazy loading)	4
Render the template on demand	4
Clear content on sse connection error	4

Reactor with spring

Linking a spring event to a Flux

Create a sink

```
// The exact config depends on the needs
Sinks.Many<Event> sink = Sinks.many().replay().limit(1);

@EventListener(Event.class)
public void onTick(Event event) {
    sink.tryEmitNext(event);
}

public Flux<Event> someMethod() {
    return sink.asFlux();
}
```

HTMX with Spring boot and flux

Configure maven

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
  </dependency>
```

```

<dependency>
  <groupId>nz.net.ultraq.thymeleaf</groupId>
  <artifactId>thymeleaf-layout-dialect</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-webflux</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>

<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <optional>true</optional>
</dependency>
<dependency>
  <groupId>org.webjars</groupId>
  <artifactId>webjars-locator</artifactId>
  <version>0.46</version>
</dependency>
<dependency>
  <groupId>org.webjars.npm</groupId>
  <artifactId>htmx.org</artifactId>
  <version>1.9.6</version>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>io.projectreactor</groupId>
  <artifactId>reactor-test</artifactId>
  <scope>test</scope>
</dependency>
</dependencies>

```

Basic template configuration

main.html

```

<!DOCTYPE html>
<html th:lang="|${#locale.language}-${#locale.country}|"
      xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout"
      xmlns:th="http://www.thymeleaf.org">
  <head>

```

```

    <meta charset="UTF-8">
    <meta content="width=device-width, initial-scale=1" name="viewport">
    <title></title>
    <!--    <link-->
    <!--          rel="stylesheet"-->
    <!--
href="https://cdn.jsdelivr.net/npm/@picocss/pico@2/css/pico.min.css"-->
    <!--    />-->

</head>

<body>
    <main layout:fragment="content">
    </main>

    <script th:src="@{/webjars/hmx.org/dist/hmx.min.js}"
type="text/javascript"></script>
    <script th:src="@{/webjars/hmx.org/dist/ext/sse.js}"
type="text/javascript"></script>
    <th:block layout:fragment="script-content">
    </th:block>
</body>
</html>

```

index.html

```

<!DOCTYPE html>
<html lang="en"
    layout:decorate="~{layout/main}"
    xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout">
    <body>
        <div layout:fragment="content">
            ... content should be here ...
        </div>
    </body>
</html>

```

Push from the backend

```

@GetMapping("/time")
public Flux<ServerSentEvent<String>> progress() {
    return timeFlux
        .map(e -> ServerSentEvent.<String>builder().data("<div>" + e +
"</div>").build());
}

```

```
<div hx-ext="sse" sse-connect="/time" sse-swap="message">
  --- This will be replaced by the response of /time ---
</div>
```

Get data at page load (lazy loading)

```
<div hx-get="/versions" hx-trigger="load">
  --- This will be replaced by the response of /time ---
</div>
```

Render the template on demand

```
import org.thymeleaf.TemplateEngine;

@Autowired
private final TemplateEngine templateEngine;

private String render(String template, Map<String, Object> parameter) {
    IContext context = new Context(Locale.getDefault(), parameter);
    return templateEngine.process(template, context).replace("\r", "").replace("\n",
    "");
}
```

Clear content on sse connection error

```
<script>
  function clearcontent(elementID) {
    document.getElementById(elementID).innerHTML = "<H1>Connection error with the
    backend!</H1>";
  }
</script>

<script>
  document.body.addEventListener('htmx:sseError', function (evt) {
    clearcontent('monitoring-sse')
  });
</script>
```