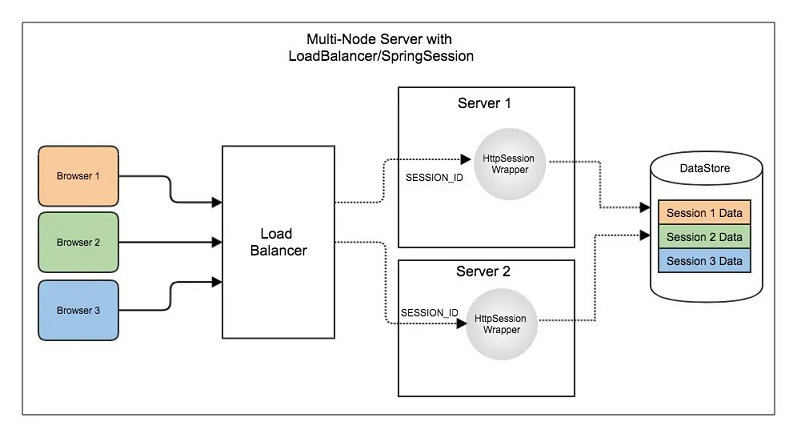
1. [RESPECTABLE]

Proof of Concept to demonstrate sharing session between different applications. Expectation is to have scalable session management where there is addition of new applications.

**Spring Session** provides the simple goal to free up session management from the limitations of the HTTP session stored in a server. This solution makes it easy to share session data between services in the cloud without being tied to a single container.



The approach is to store session data in external storage container like DB or in Redis server.

For this POC, REDIS is used to store the session data instead of using HttpSession. Spring Session replaces the HttpSession with an implementation backed by Redis.

Below are the redis dependencies added to project:

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-session-data-redis</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-redis</artifactId>

</dependency>

**GIT REPO:**

[**https://github.com/tinaash/SessionManagement.git**](https://github.com/tinaash/SessionManagement.git)

[**git@github.com:tinaash/SessionManagement.git**](mailto:git@github.com:tinaash/SessionManagement.git)

**IMPLEMENTATION DETAILS:**

For this POC, session data is being stored and retrieved in the same Spring application, but this approach can be improved to share session data among multiple applications.

Configuring connectivity from Spring Session to Redis is done by adding the following configuration settings to the application.properties file:

spring.session.store-type=redis

spring.redis.host=localhost

spring.redis.port=6379

The configuration spring.session.store-type=redis is equivalent to manually adding @EnableRedisHttpSession annotation.

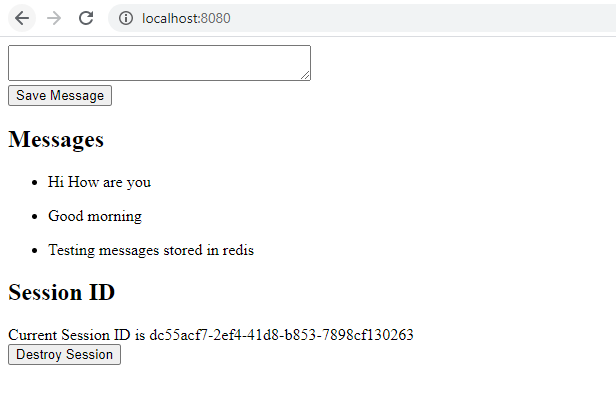
Spring Boot automatically creates a RedisConnectionFactory that connects Spring Session to a Redis Server on localhost on port 6379.

**TEST RESULTS:**

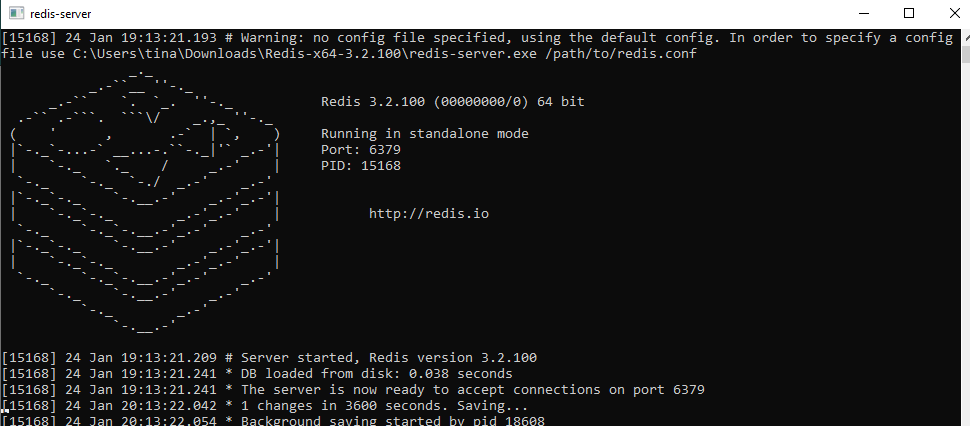
Index page to enter message as user input.

This message is stored in the redis session data





**REDIS running in default port 6379:**



**Session Data stored in Redis:**

