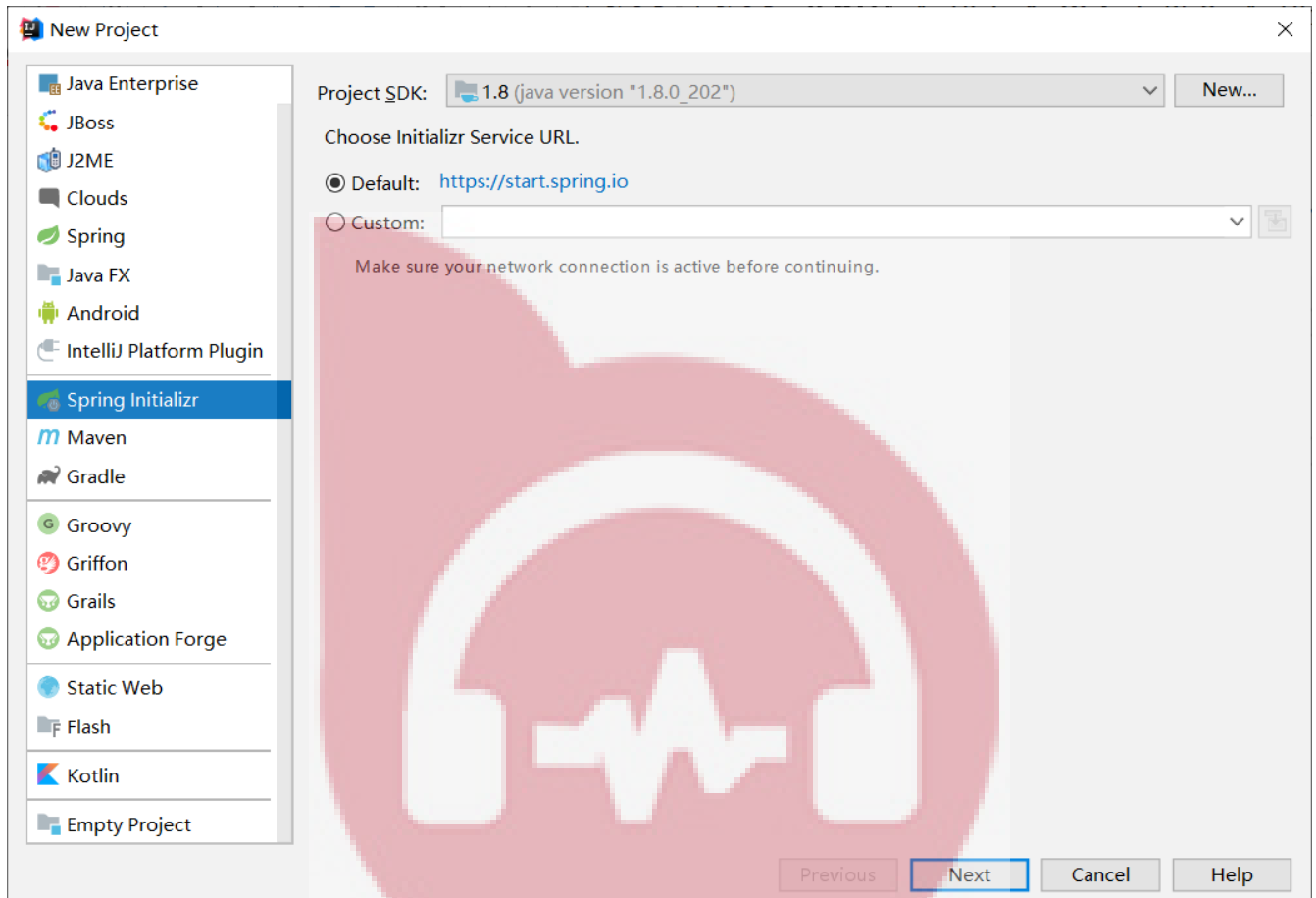


## Java操作Redis

### 创建项目

#### 创建项目





New Project

Project Metadata

Group:

com.xxxx

Artifact:

redisdemo

Type:

Maven Project (Generate a Maven based project archive.)

Language:

Java

Packaging:

Jar

Java Version:

8

Version:

0.0.1-SNAPSHOT

Name:

redisdemo

Description:

Demo project for Spring Boot

Package:

com.xxxx.redisdemo

Previous

Next

Cancel

Help

New Project

Dependencies

Spring Boot 2.2.4

Developer Tools

Web

Template Engines

Security

SQL

NoSQL

Messaging

I/O

Ops

Testing

Spring Cloud

Spring Cloud Security

Spring Cloud Tools

Spring Cloud Config

Spring Cloud Discovery

Spring Cloud Routing

Spring Cloud Circuit Breaker

Spring Cloud Tracing

Spring Cloud Messaging

Pivotal Cloud Foundry

Amazon Web Services

Microsoft Azure

☒ Spring Data Redis (Access+Driver)

☐ Spring Data Reactive Redis

☐ Spring Data MongoDB

☐ Spring Data Reactive MongoDB

☐ Spring Data Elasticsearch (Access+Driver)

☐ Spring Data for Apache Solr

☐ Spring Data for Apache Cassandra

☐ Spring Data Reactive for Apache Cassandra

☐ Spring for Apache Geode

☐ Spring Data Couchbase

☐ Spring Data Reactive Couchbase

☐ Spring Data Neo4j

Spring Data Redis (Access+Driver)

Advanced and thread-safe Java Redis client for synchronous, asynchronous, and reactive usage. Supports Cluster, Sentinel, Pipelining, Auto-Reconnect, Codecs and much more.

🏠 Messaging with Redis

🔍 Reference doc

Selected Dependencies

Web

Spring Web

NoSQL

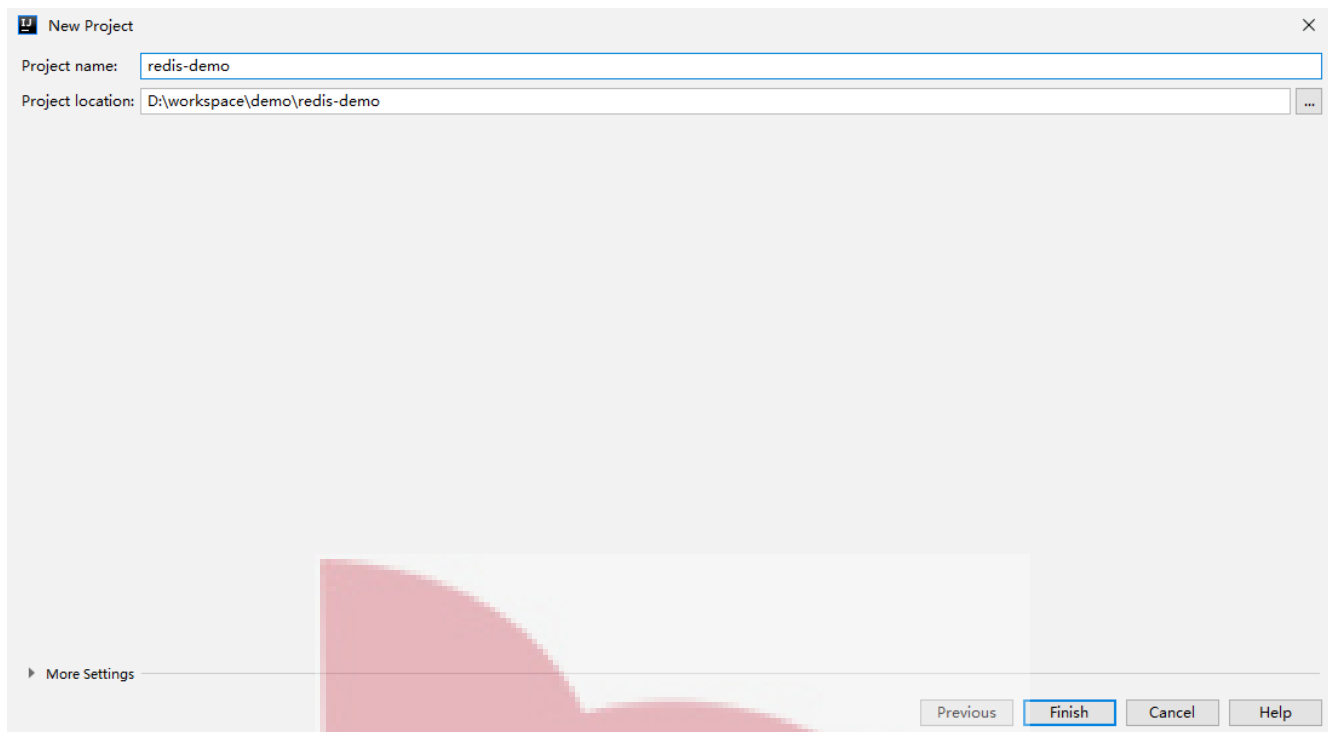
Spring Data Redis (Access+Driver)

Previous

Next

Cancel

Help



## 添加依赖

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.2.4.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>

  <groupId>com.xxxx</groupId>
  <artifactId>redisdemo</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>redisdemo</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <!-- spring data redis 组件 -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-redis</artifactId>
    <!--
      1.x 的版本默认采用的连接池技术是 Jedis,
```



2.0 以上版本默认连接池是 Lettuce，  
如果采用 Jedis，需要排除 Lettuce 的依赖。

```
-->
<exclusions>
  <exclusion>
    <groupId>io.lettuce</groupId>
    <artifactId>lettuce-core</artifactId>
  </exclusion>
</exclusions>
</dependency>
<!-- jedis 依赖 -->
<dependency>
  <groupId>redis.clients</groupId>
  <artifactId>jedis</artifactId>
</dependency>
<!-- web 组件 -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<!-- test 组件 -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
</dependencies>
</project>
```

## 配置文件

```
spring:
  redis:
    # Redis服务器地址
    host: 192.168.10.100
    # Redis服务器端口
    port: 6379
    # Redis服务器密码
    password: root
    # 选择哪个库，默认0库
    database: 0
    # 连接超时时间
    timeout: 10000ms
  jedis:
    pool:
      # 最大连接数，默认8
      max-active: 1024
      # 最大连接阻塞等待时间，单位毫秒，默认-1ms
      max-wait: 10000ms
      # 最大空闲连接，默认8
      max-idle: 200
      # 最小空闲连接，默认0
      min-idle: 5
```



## Java怎么连接Redis?

```
/**
 * 连接Redis
 */
@Test
public void initConn01() {
    // 创建jedis对象, 连接redis服务
    Jedis jedis = new Jedis("192.168.10.100", 6379);

    // 设置认证密码
    jedis.auth("root");

    // 指定数据库 默认是0
    jedis.select(1);

    // 使用ping命令, 测试连接是否成功
    String result = jedis.ping();
    System.out.println(result); // 返回PONG

    // 添加一条数据
    jedis.set("username", "zhangsan");

    // 获取一条数据
    String username = jedis.get("username");
    System.out.println(username);

    // 释放资源
    if (jedis != null)
        jedis.close();
}
```