好的，这是一个非常好的问题，很多初学者都会对这几个名字相似的依赖感到困惑。它们之间有明确的区别和演进关系。

根据您截图中的信息，我们来详细分析一下。

### **快速回答**

当我们在现代的 Spring Cloud 项目中（一般指基于 Spring Boot 2.x）谈论添加 Zuul 依赖时，我们**通常指的是第一个：spring-cloud-starter-netflix-zuul**。这是你现在应该在 pom.xml 中使用的唯一正确的启动器。

### **详细分析：三个依赖的角色**

为了方便理解，我们可以把它们比作“汽车套装”、“汽车发动机”和“旧款汽车套装”。

| 依赖 (Dependency) | 角色与作用 | 何时使用 |
| --- | --- | --- |
| 🥇 **spring-cloud-starter-netflix-zuul** | **一站式启动器 (The "All-in-One" Starter)** | **在所有现代的、基于 Spring Boot 2.x 的 Zuul 网关项目中，你应该只使用这个依赖。** |
|  | 这是 Spring Boot 的标准“启动器”模式。它本身不包含太多代码，主要作用是作为一个**依赖描述符**，自动地、可传递地将构建 Zuul 网关所需的**所有相关库**（包括核心实现、自动配置、熔断、服务发现等）一次性引入到你的项目中。**这是最省心、最正确的选择。** |  |
| ⚙️ **spring-cloud-netflix-zuul** | **核心实现库 (The "Core" Library)** | **通常情况下，你永远不需要直接将它添加到你的 pom.xml 中。** |
|  | 这是 Zuul 功能的**核心引擎**。它包含了 @EnableZuulProxy 注解的实现、与 Eureka 客户端集成的逻辑、以及 ZuulFilter 等核心类。上面的 starter 会自动把它作为依赖引入。 | 只有在你需要极其精细的控制，希望使用 Zuul 的核心功能但又想**禁用 Spring Boot 的自动配置**时，才可能会直接依赖它。这种情况非常罕见。 |
| 🚫 **spring-cloud-starter-zuul** | **已废弃的旧版启动器 (The Deprecated/Old Starter)** | **永远不要在任何新项目中使用它。** |
|  | 这是在**早期 Spring Cloud 版本**（对应 Spring Boot 1.x）中使用的启动器名称。正如您截图中的描述文字 (deprecated, please use spring-cloud-starter-netflix-zuul) 所示，它已经被明确**废弃**。在后来的版本中，为了更好地组织来自 Netflix OSS 的组件，Spring Cloud 在artifactId中加入了 -netflix-。 | 你只会在维护非常古老的、基于 Spring Boot 1.x 的遗留项目中看到它。 |

### **总结与关系图**

这三者之间的关系可以用下图来表示：

你的项目 pom.xml  
 │  
 └───> 依赖于: spring-cloud-starter-netflix-zuul (✅ 你应该添加这个)  
 │  
 └───> 自动引入: spring-cloud-netflix-zuul (核心库)  
 │  
 └───> 自动引入: 其他所需库 (如 Ribbon, Hystrix-javanica 等)

**简单来说：**

1. **你要用** -> spring-cloud-starter-netflix-zuul
2. **它帮你用** -> spring-cloud-netflix-zuul
3. **不要用** -> spring-cloud-starter-zuul (除非是维护比石器时代还老的项目)

**最后再次提醒**：虽然我们讨论的是 Zuul 的依赖，但 Spring Cloud Zuul 1 已经进入维护模式。对于所有新项目，官方都强烈推荐使用其继任者 **Spring Cloud Gateway**。