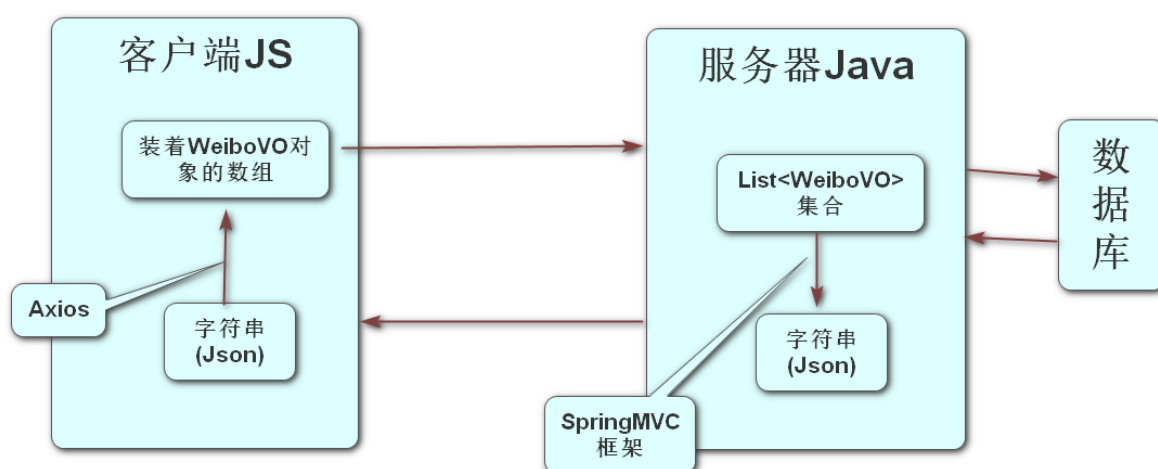


## 同步请求和异步请求

- 同步:指单线程依次做几件事
- 异步:指多线程同时做几件事
- 同步请求:指客户端只有一个主线程,主线程既要负责页面渲染\监听用户操作,还需要负责发请求,当主线程发请求时,会将页面内容清空,直到服务器响应了数据之后再将响应的数据展示到页面中,这种页面整体改变称为页面整体刷新,同步请求是无法实现页面的局部刷新的.
- 异步请求: 指客户端浏览器主线程只负责页面渲染\监听用户操作,由子线程负责发出请求,当子线程请求到数据后,可以将数据展示到原页面中,这就是页面的局部刷新.

## 客户端和服务端之间数据交互过程



## Security框架认证流程:

1. 在pom.xml里面添加Security框架的依赖, 然后刷新Maven, 此时工程访问任何资源都会跳转到Security框架自带的登录页面

```
<!-- Spring Boot支持Spring Security的依赖项, 用于处理认证与授权 -->
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

2. 创建Security框架的配置类,重写configure方法并删除掉调用父类方法的代码, 在里面配置了自己的登录页面,设置白名单,关闭跨域攻击防御

```
@Slf4j
@Configuration
public class SecurityConfig extends WebSecurityConfigurerAdapter {
```

```

@Override
protected void configure(HttpSecurity http) throws Exception {
    //super.configure(http);
    //配置自己的登录页面 当判断没有登录的时候 自动跳转到自己的登录页面
    http.formLogin().loginPage("/login.html");

    //配置白名单(无需登录也可以访问的资源)
    String[] urls = {"/reg.html", "/login.html", "/reg", "/login"};
    http.authorizeRequests()
        .mvcMatchers(urls)//匹配某些路径
        .permitAll() //直接放行(不需要登录可以访问)
        .anyRequest()//其它请求
        .authenticated(); //需要登录认证才能访问

    //关闭默认开启的跨域攻击防御
    http.csrf().disable();
}
}

```

3. 创建了UserDetailServiceImpl.java这是UserDetailsService接口的实现类, 在里面实现了loadUserByUsername方法, 此方法是当开启Security框架的认证时自动调用的方法, 此方法里面如果return出的是一个null代表用户输入的用户名是不存在的, 如果想要出的是一个UserDetails对象代表用户名存在, 密码是否正确由框架内部的代码进行判断.

```

@Override //此方法的username代表用户输入的用户名
public UserDetails loadUserByUsername(String username) throws
UsernameNotFoundException {
    //此方法当Security框架进行登录认证时, 自动调用
    //当此方法响应null时 代表用户名不存在, 下面模拟用户输入的用户名是没问题的
    //假设tom和123456是从数据库里面查询出来的数据
    if (username.equals("tom")){
        UserDetails userDetails = User.builder()
            .username("tom").password("123456")
            .disabled(false)//账号是否禁用
            .accountLocked(false)//账号是否锁定
            .accountExpired(false)//账号是否过期
            .credentialsExpired(false)//登录凭证是否过期
            .authorities("权限名") //授予当前登录的用户的权限
            .build();
        return userDetails;
    }
    return null;//代表用户名不存在
}
}

```