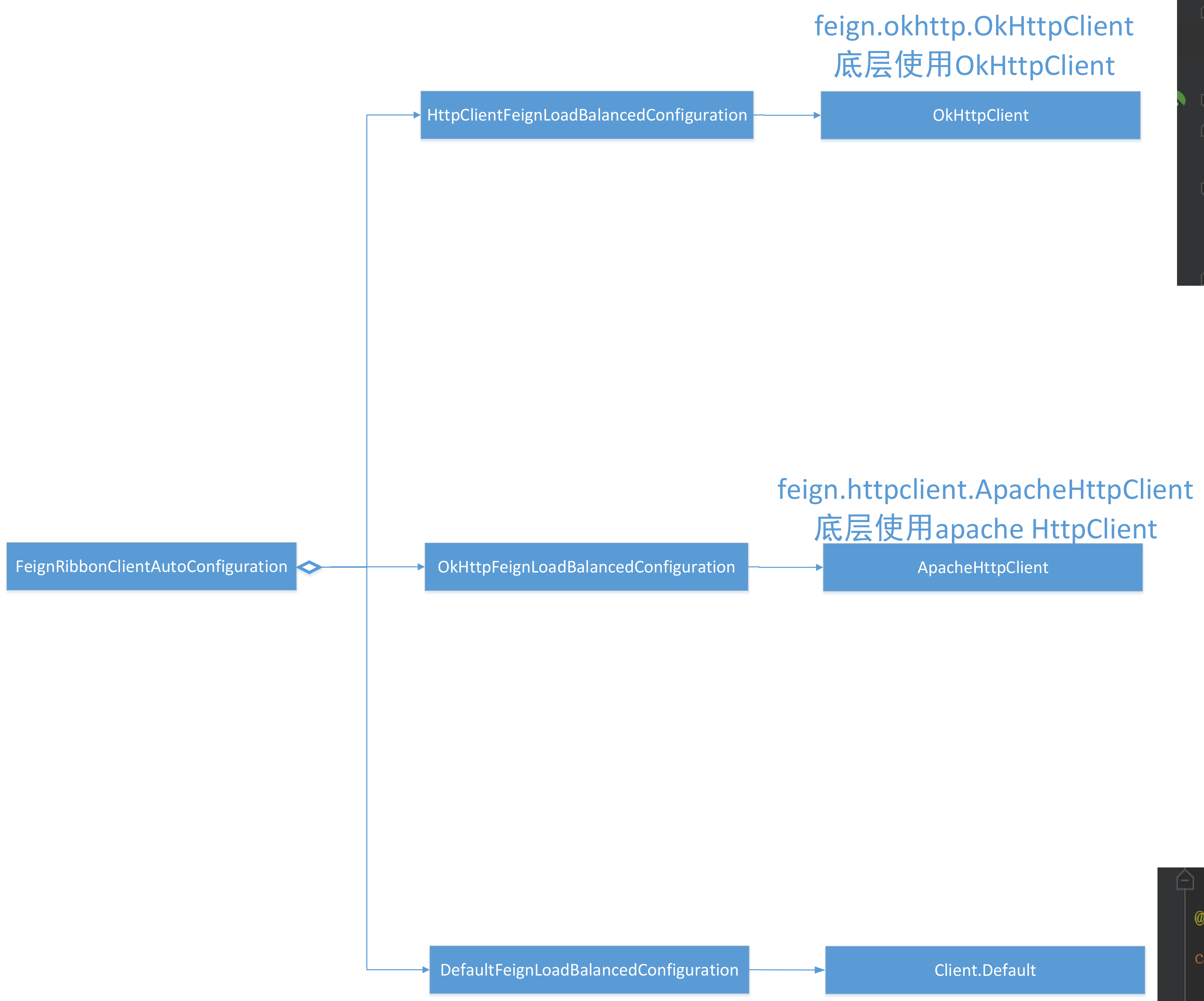
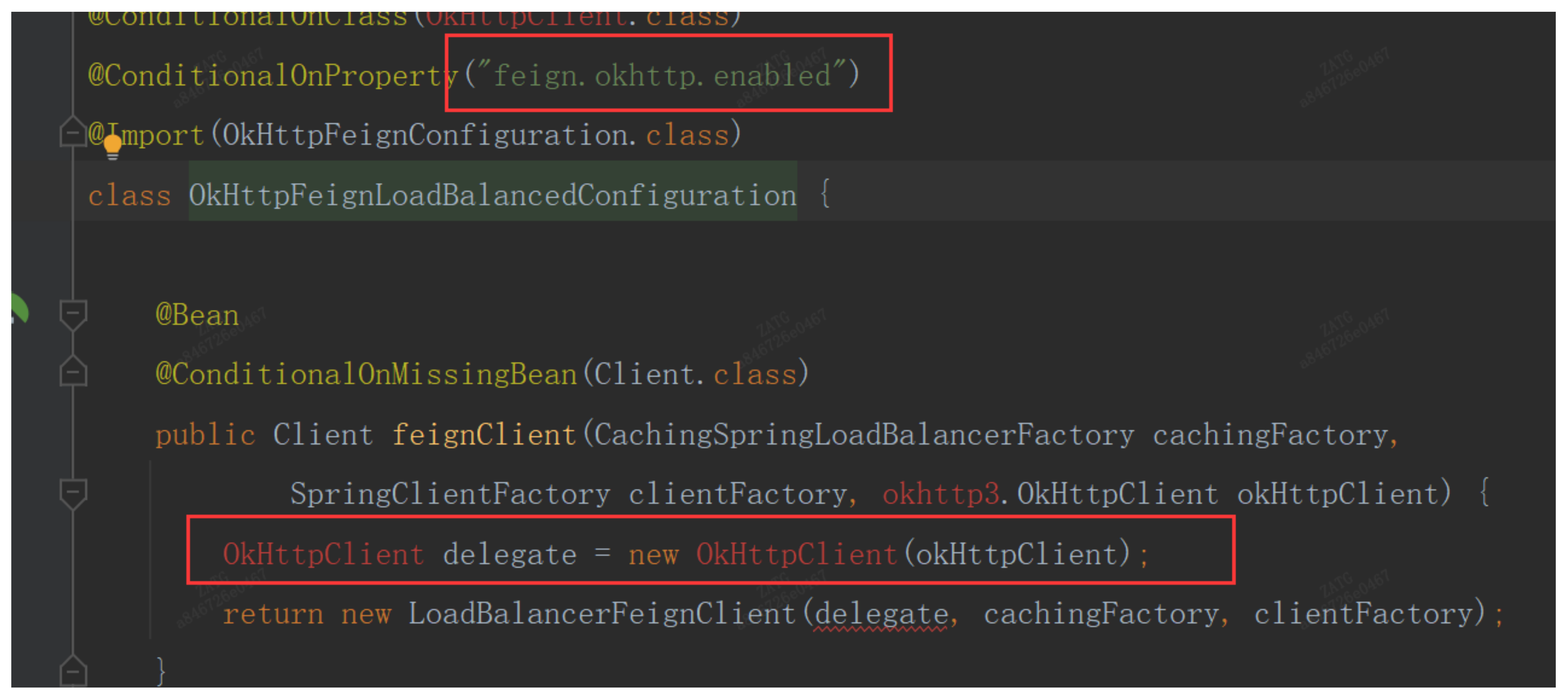
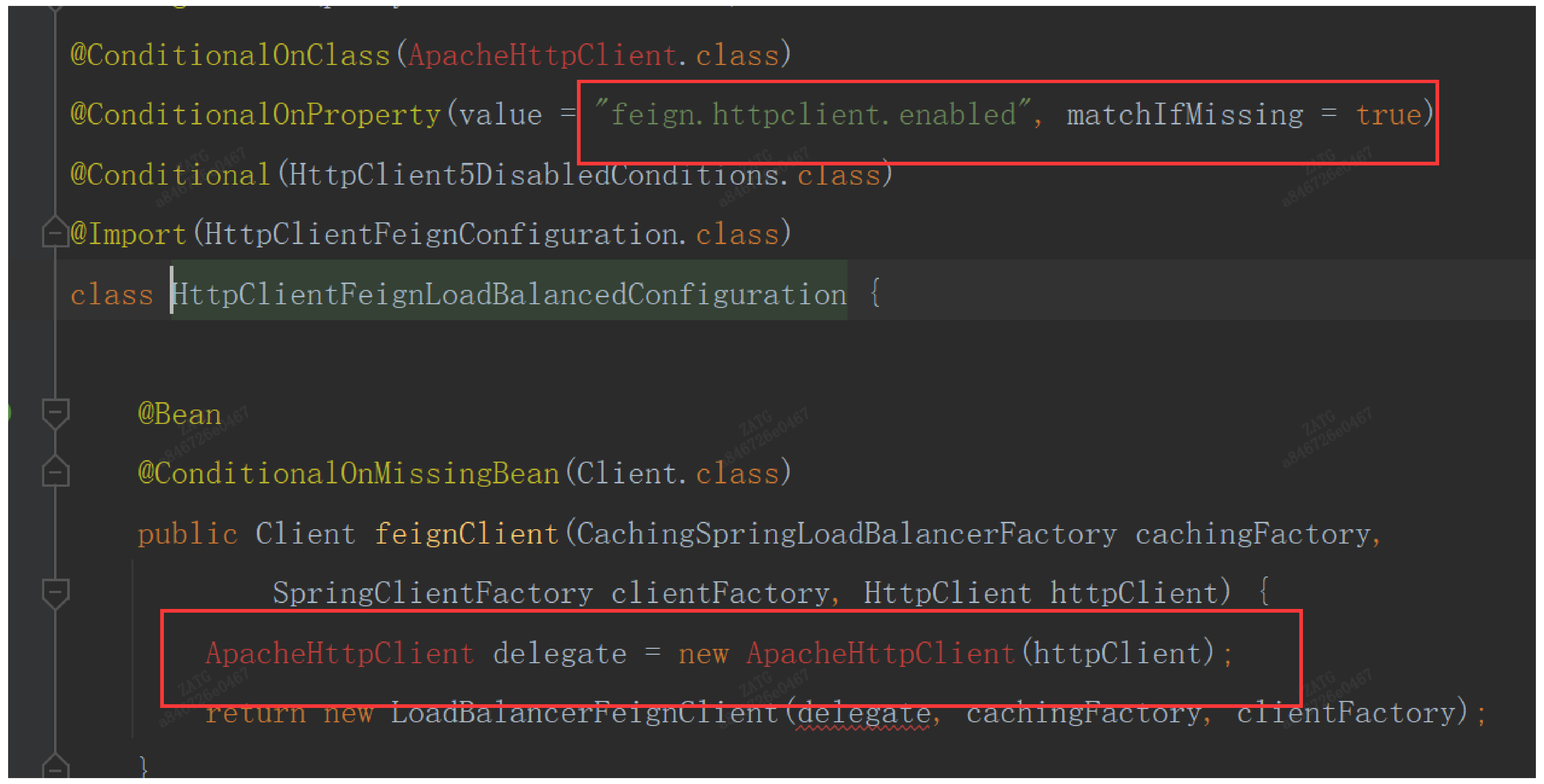

feign.okhttp.OkHttpClient
底层使用OkHttpClient
feign.httpclient.ApacheHttpClient
底层使用apache HttpClient
FeignRibbonClientAutoConfiguration
HttpClientFeignLoadBalancedConfiguration
OkHttpClient
OkHttpFeignLoadBalancedConfiguration
ApacheHttpClient
DefaultFeignLoadBalancedConfiguration
Client.Default



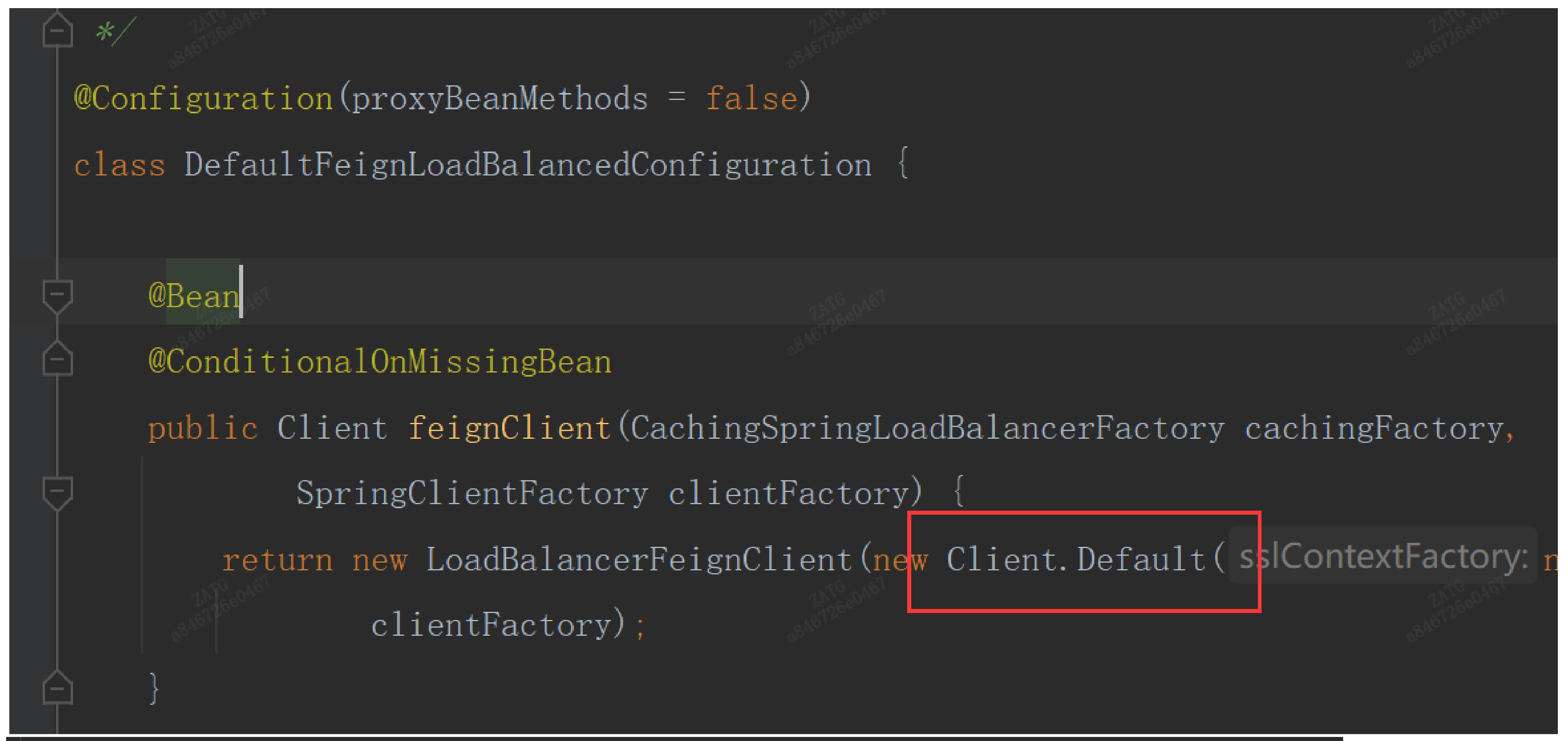
```
@ConditionalOnClass(OkHttpClient.class)
@ConditionalOnProperty("feign.okhttp.enabled")
@Import(OkHttpFeignConfiguration.class)
class OkHttpFeignLoadBalancedConfiguration {

    @Bean
    @ConditionalOnMissingBean(Client.class)
    public Client feignClient(CachingSpringLoadBalancerFactory cachingFactory,
            SpringClientFactory clientFactory, okhttp3.OkHttpClient okHttpClient) {
        OkHttpClient delegate = new OkHttpClient(okHttpClient);
        return new LoadBalancerFeignClient(delegate, cachingFactory, clientFactory);
    }
```



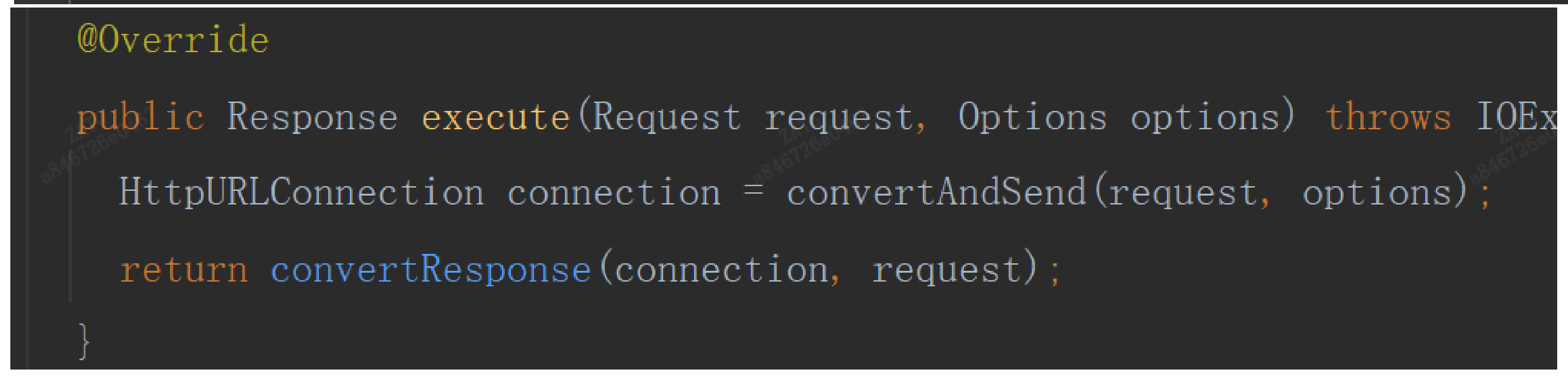
```
@ConditionalOnClass(ApacheHttpClient.class)
@ConditionalOnProperty(value = "feign.httpclient.enabled", matchIfMissing = true)
@Conditional(HttpClient5DisabledConditions.class)
@Import(HttpClientFeignConfiguration.class)
class HttpClientFeignLoadBalancedConfiguration {

    @Bean
    @ConditionalOnMissingBean(Client.class)
    public Client feignClient(CachingSpringLoadBalancerFactory cachingFactory,
            SpringClientFactory clientFactory, HttpClient httpClient) {
        ApacheHttpClient delegate = new ApacheHttpClient(httpClient);
        return new LoadBalancerFeignClient(delegate, cachingFactory, clientFactory);
    }
```



```
 */
@Configuration(proxyBeanMethods = false)
class DefaultFeignLoadBalancedConfiguration {

    @Bean
    @ConditionalOnMissingBean
    public Client feignClient(CachingSpringLoadBalancerFactory cachingFactory,
            SpringClientFactory clientFactory) {
        return new LoadBalancerFeignClient(new Client.Default( sslContextFactory: n
                clientFactory);
    }
```



```
@Override
public Response execute(Request request, Options options) throws IOEx
  HttpURLConnection connection = convertAndSend(request, options);
  return convertResponse(connection, request);
}
```