

```

    } else if (304 == result.code) {
        return "";
    } else {
        throw new NacosException(500, "failed to req API:" +
HttpClient.getPrefix() + curServer + api + ". code:" + result.code + " msg: "
+ result.content);
    }
}

```

上面的代码就是将注册请求发送到 Nacos Server。

接着, Nacos 开始接收请求并处理请求。

Nacos 服务注册 controller 中的 register()方法的代码如下。

```

@CanDistro
@PostMapping
public String register(HttpServletRequest request) throws Exception {
    String serviceName = WebUtils.required(request,
CommonParams.SERVICE_NAME);
    String namespaceId = WebUtils.optional(request,
CommonParams.NAMESPACE_ID, Constants.DEFAULT_NAMESPACE_ID);
    serviceManager.registerInstance(namespaceId, serviceName,
parseInstance(request));
    return "ok";
}

```

最后来看一下 ServerManager 的具体实现。

```

public void registerInstance(String namespaceId, String serviceName,
Instance instance) throws NacosException {
    createEmptyService(namespaceId, serviceName,
instance.isEphemeral());
    Service service = getService(namespaceId, serviceName);
    if (service == null) {
        throw new NacosException(NacosException.INVALID_PARAM,
"service not found, namespace: " + namespaceId + ", service:
" + serviceName);
    }
    addInstance(namespaceId, serviceName, instance.isEphemeral(),
instance);
}

```

从以上 Nacos 源码可以了解 Nacos 的服务注册原理。