## HTTP Verbs：

1. 是否使用X-HTTP-Method-Override？  
    X-HTTP-Method-Override是HTTP Verbs的一种渐进式的兼容方案，针对部分不支持PUT、PATCH、HEAD的浏览器、服务器，统一采用POST方法发送请求，在X-HTTP-Method-Override中添加对应的verb。
2. 是否使用PUT、PATCH区别更新？
3. 是否采用HEAD？  
    一般使用HEAD进行资源的检查操作。如检查某用户是否收藏了某书，/book/{bookId}/collection{?userId}，如果用GET，语义是返回对应这条收藏的数据库记录的实体。而客户端的需求仅仅是检测是否存在，而这个结果通过检查response status code是200还是404就可以得到。所以这里根本不需要使用GET、使用HEAD可以实现，节省服务器带宽。

* 使用X-HTTP-Method-Override
* 不使用X-HTTP-Method-Override
* GET：获取资源
* POST：创建资源
* PUT：修改资源
* PATCH：修改部分资源
* DELETE：删除资源
* HEAD：检测资源状态

## HTTP status code：

1. 使用HTTP status code表达操作结果，业务错误放在500返回中，权限错误放在401返回中，统一采用JSON{code: ‘’; msg: ‘’}格式
2. 统一采用200返回，使用JSON表达操作结果，JSON采用{code: ‘’;msg:’’;data:[]/{}}格式

## RESTful化后的登录流程实例（护理质量）：

1. 客户端进入web app，进行系统的初始化，包括获取版本号、获取系统配置项等
2. 客户端同时检查用户状态，如没有用户状态，跳转登录页面
3. 客户端从登录页面向/token发送POST请求
4. 服务器收到请求，生成token，返回token的标识符key和用户的标识符id
5. 客户端收到返回，将token绑定到之后的request的Authorization头部中，将id作为用户最初始的状态，跳转页面到首页，并发送login事件供监听函数统一处理
6. 客户端监听函数收到事件，进行登录后信息的初始化
7. 客户端根据业务需求，发送/user/:userId获取基本信息
8. 服务器收到请求，根据token对应的userId和path中请求的userId做匹配，确定数据的所有人与令牌的所有人的权限关系，根据权限管理原则，做出是否返回数据的抉择，分别对应200和403。如果返回数据，根据信息分层原则返回对应的字段组成的数据。由于这里是申请登录人员本人的基本信息，返回所有数据
9. 客户端收到返回，对客户端用户状态做出更新，状态由id更新为人员基本信息，缓存在model层供其他组件调用，并在view层提供一个指向model层的引用，供基础视图组件如header显示调用。
10. 客户端以人员基本信息作为基础，根据业务需求，初始化更多的个人信息。比如角色对应的菜单信息，发送/role/:roleId/menu
11. 服务器收到请求，这里根据权限管理原则，决定是否返回menu。正常流程肯定是返回menu的
12. 客户端收到menu，缓存在view层供navbar显示调用。

## API URL的基本格式：

http://{host}:{port}/{contextPath}/api/{version}{/resource \*}

GET的方式

1. http://{host}:{port}/{contextPath}/api/{version}{/resource \*}{?offset,num,order,filters}
2. http://{host}:{port}/{contextPath}/api/{version}{/resource \*}{?offset,num,order}{&filters \*}
3. http://{host}:{port}/{contextPath}/api/{version}{/resource \*}{?offset,num,order}{&filters}

版本号的获取方式

1. 系统首选项：http://{host}:{port}/{contextPath}/api
2. 前端配置常量

## 后端对跨域访问的支持：

Access-Control-Allow-Origin: \*

Access-Control-Allow-Methods:GET, POST, PUT, DELETE, PATCH

Access-Control-Allow-Headers: Authorization

共用的返回字段、请求参数、标识符  
返回字段

createPerson：创建人

createTime：创建时间

updatePerson：更新人

updateTime：更新时间

时间的格式

1. 时间字符串：如之前的2016-12-12 12:12:12
2. 时间戳：如毫秒级的1478858668360，秒级的1478858668
3. UTC本地化字符串：Mon Nov 14 2016 09:22:02 GMT+0800

请求参数

offset：偏移量

num：数量

order：排序方式  
filter的请求参数建议直接用对应model的field本身  
  
标识符  
model不同而不同，如科室的deptCode，用户的userCode，建议统一用moduleName + Code形式