

REST API

- Rest API 简介
- Restful authentication
- My veivo rest api design
- JSR311
- Jersey
- Jersey Demo

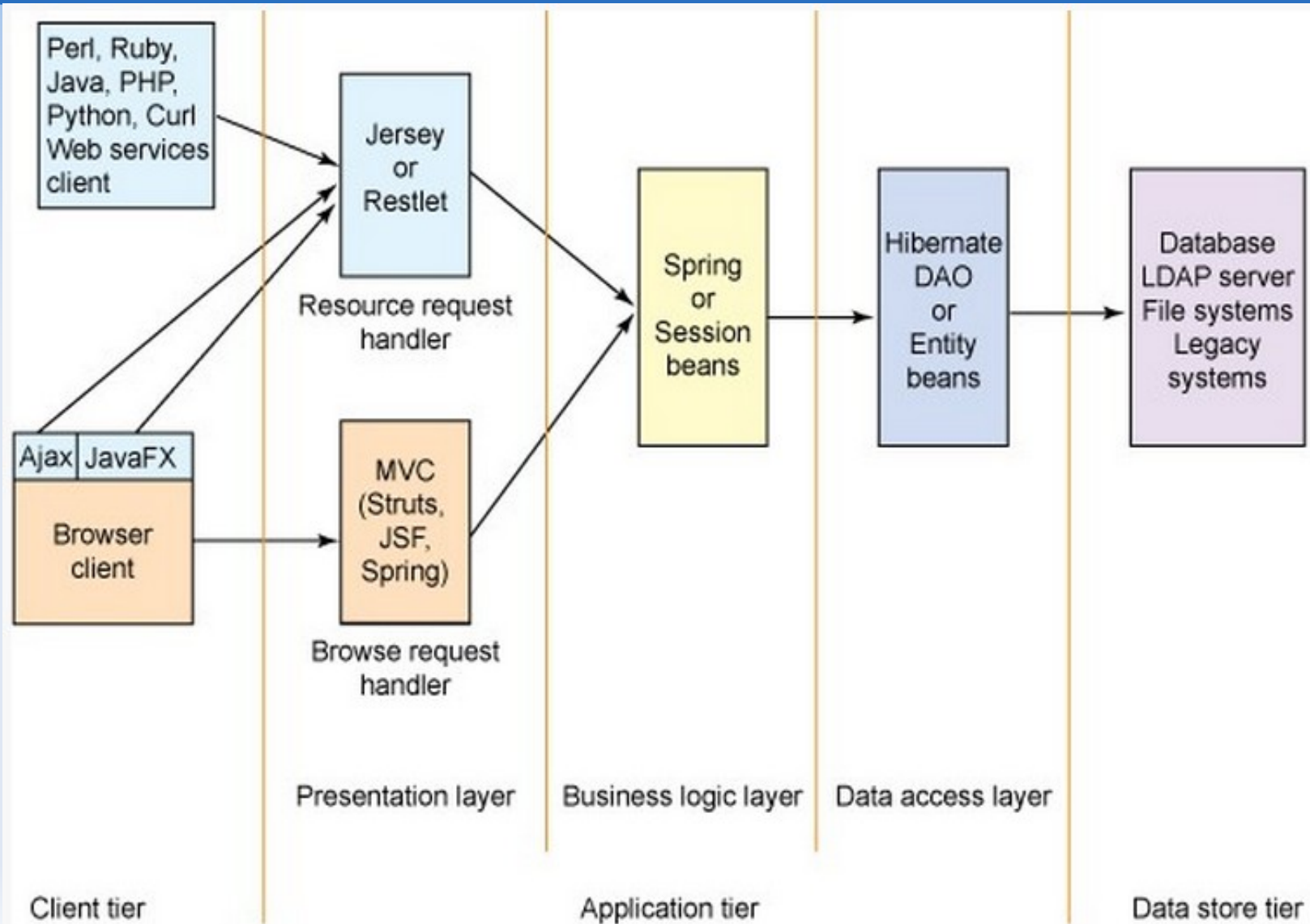
REST API

- Rest API 简介

REST 指的是一组架构约束条件和原则。

- 客户端和服务端之间的交互在请求之间是无状态的。服务端不保存任何客户端信息，便于分布式处理。
- 另一个重要的 REST 原则是分层系统，降低复杂性。

REST API



Restful Authentication

Amazon:

- RESTful API Authentication Schemes
- George Reese 在文中概括了他认为 REST API Authentication 所应该遵循的 3 条原则：

Restful Authentication

- 1. All REST API calls must take place over **HTTPS** with a certificate signed by a trusted CA. All clients must validate the certificate before interacting with the server.
- 所有 REST API 请求都必须通过加密的 HTTPS 协议来传递，加密所用的证书应由一个 trusted CA (certificate authority) 签发。所有客户端必须验证服务器端的身份证书，然后才开始进行会话。

Restful Authentication

2. 所有 REST API 必须由专门的 API 密钥来验证。API 密钥 (注意: 这里的密钥是广义的, 不一定是指 public key authentication 中的 key) 必须有一个身份确认段, 以及一个客户端 / 服务器端共享的私有秘密。服务器端的系统必须允许每个用户拥有一个或多个 API 密钥, 并且可以方便地禁用某些密钥 (使之不再有效)。

Restful Authentication

Amazon 设计：

- 客户可以随时申请任意个 authentication token，Amazon 会通过一个 HTTPS 通道将 authentication token 发送给客户

Restful Authentication

- AuthenticationToken =
RealmString ":" TimeStamp ":" Domain ":" SignedPackage
- SignedPackage :
- Base64(S(Gpri, RealmString + ":" +
- TimeStamp + ":" + DNS + OwnerPUID))

Restful Authentication

- 3. 所有 REST 的查询请求都必须经过验证，验证的方式是客户端需对所有查询参数以小写字母顺序排序后用上文所提到的 shared, private secret 进行签名加密。签名加密的步骤应该在对查询字符串进行 URL 编码之前执行。
- 对于查询字符串加密会造成很多问题：比如所有的代理服务器的功能都无效了，比如服务器端缓存就无法实现了，最搞笑的是用户如果想把某个 REST 查询在 Facebook 上分享给朋友都不可能了。

Restful Authentication

- REST 协议的一个基本原则就是低耦合，也就是专项专用。URL 就是用来决定 resource 的位置的，而不应 该也不必要再夹带其他功能，比如身份验证。身份验证的信息完全可以通过其他标准的 HTTP 协议的组件来实现。

Veivo Rest API Design

针对一个feed，
使用HTTP GET，
获取照片集合



针对一个entry，
使用HTTP GET，获
取某个照片



针对一个entry，使
用HTTP DELETE，删
除某个照片



针对一个feed或者
entry，使用HTTP
POST，添加一张照片



针对一个entry，
使用HTTP PUT，更
新一张照片

Veivo Rest API Design

Veivo resource:

- 用户信息
- 联系人
- 消息
- 应用

Veivo Rest API Design

- Example:

- Rest API:

GET <http://veivo.com/userinfo/vincent/entry>

- {
 userid : 123456,
 username : vincent,
 fname : li
 lname : wenzhu
 email : li_wenzhu@126.com
}

Veivo Rest API Design

- Rest api 构成
- `http://veivo.com/resources/feedName/{location}?filter=.....`

参数：

SortOrder:asc desc

format:xml/json

SortField

Page

Pagesize

StartIndex

count

Veivo Rest API Design

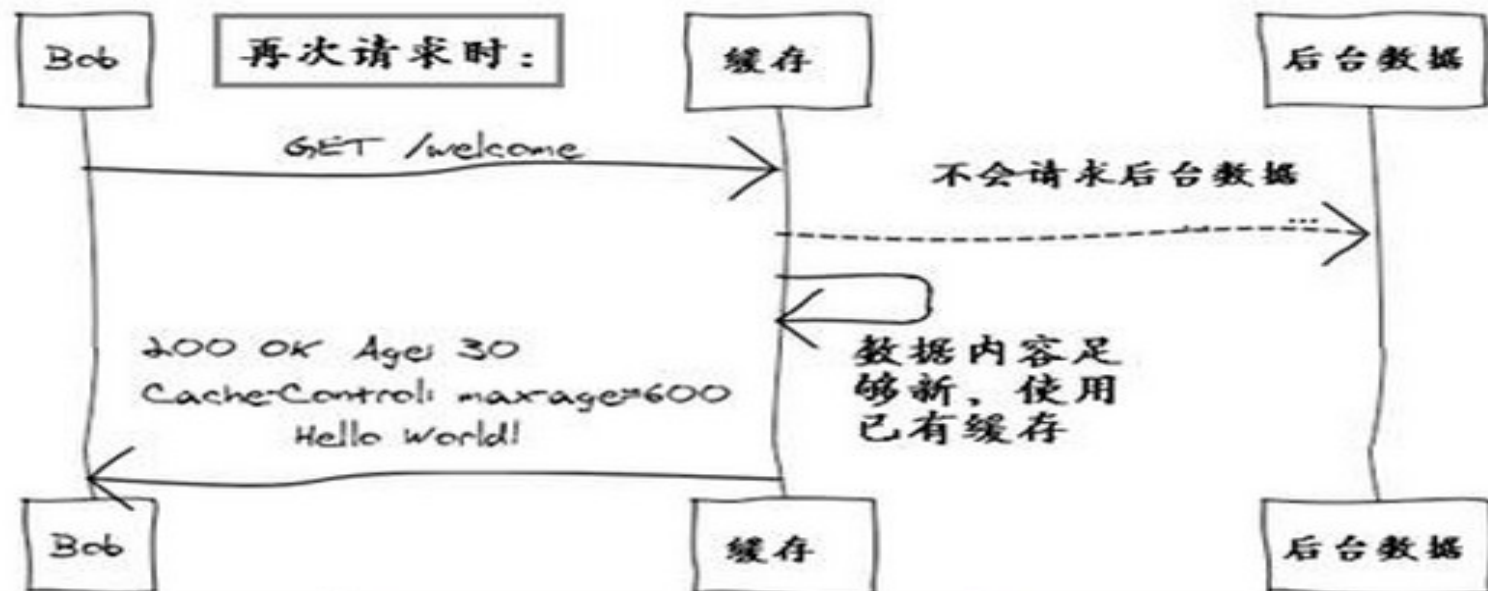
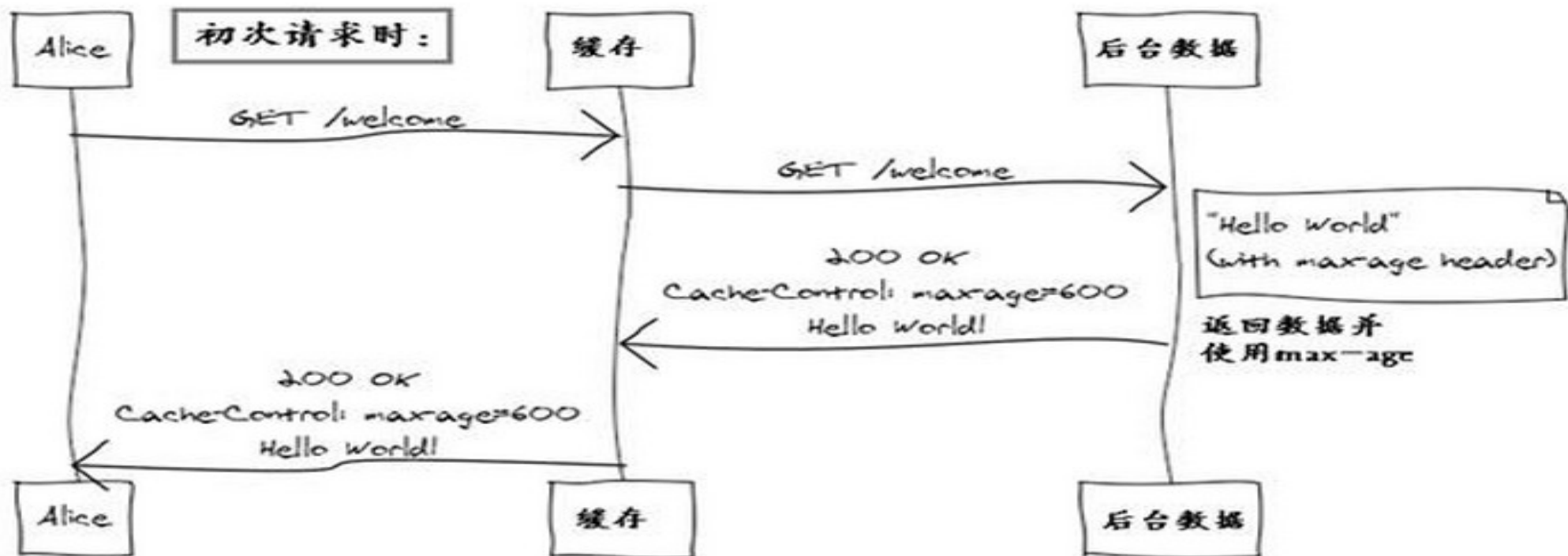
- Accept 支持

使用 URL 参数，简单灵活，但是也由此带来了设计上的随意和不标准。并且，过多的参数会导致 URL 的**可读性变差**，更有甚者，可能会导致 URL 过长，超出规范，API 请求**无法执行**。

Veivo Rest API Design

- 使用 HTTP 头处理缓存

针对 REST API 请求都是在远端用 HTTP 发起这一特点，通过网络缓存可以减少 HTTP 响应内容，避免不必要的 HTTP 连接等方式，提高 REST API 使用效率。

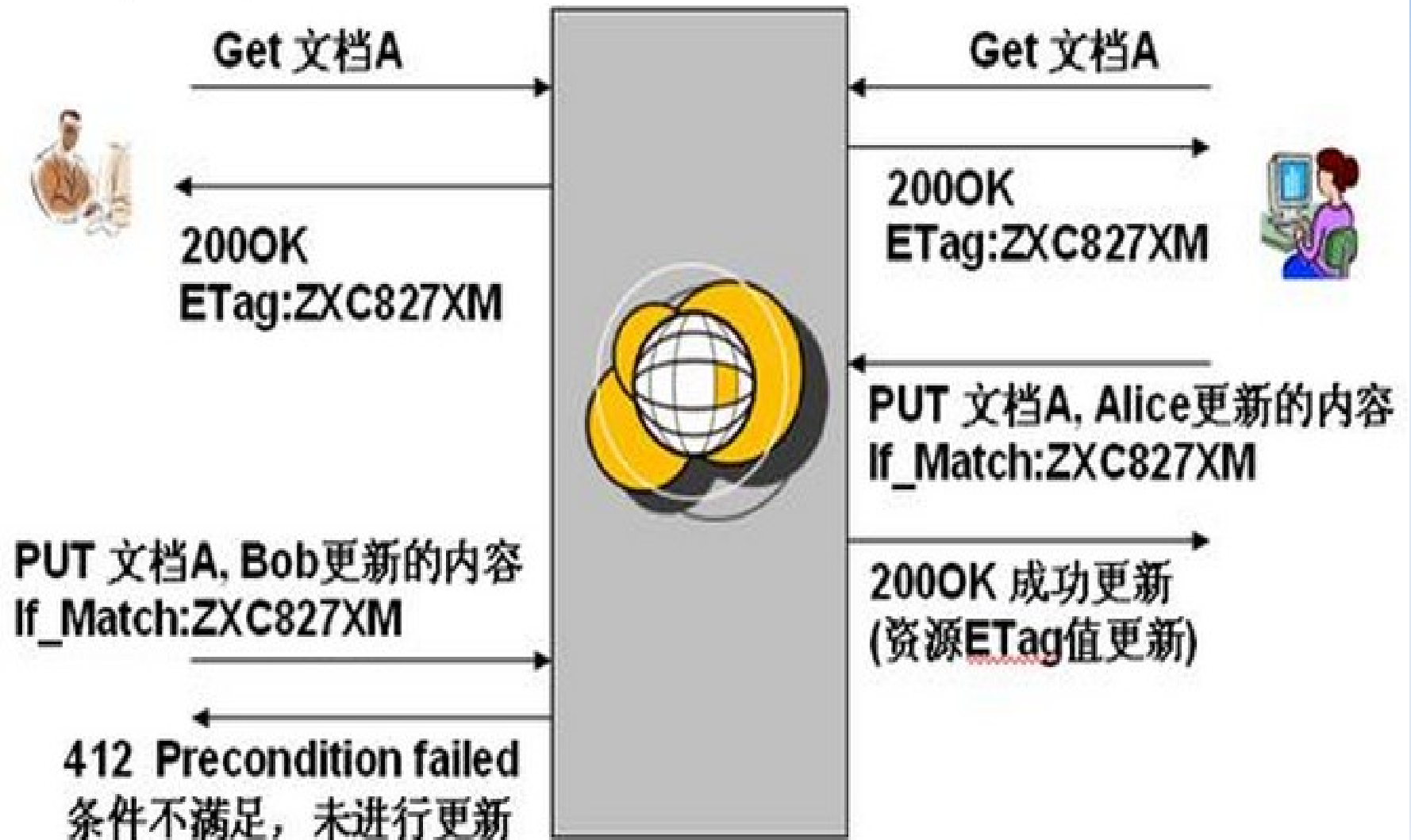


Veivo Rest API Design

- 使用 HTTP 头处理并发

在 http 请求中加入 if_match

Veivo Rest API Design



JSR311

- JSR311 正式称呼是 JAX-RS:
Java API for Restful Web Services
- JSR311 是 SUN 发布的一个基于 annotation 的 API。

JSR311

- 规范内容：
- JAX-RS 提供了一些标注将一个资源类，一个 POJO Java 类，封装为 Web 资源。

JSR311

- @Path , 标注资源类或方法的相对路径
- @GET , @PUT , @POST , @DELETE , 标注方法是用的 HTTP 请求的类型
- @Produces , 标注返回的 MIME 媒体类型
- @Consumes , 标注可接受请求的 MIME 媒体类型

JSR311

- @PathParam , @QueryParam , @HeaderParam , @CookieParam , @MatrixParam , @FormParam, 分别标注方法的参数来自于 HTTP 请求的不同位置。

Jersey

- Jersey 是由 Sun 开发的产品级质量的 JSR 311 的参考实现。 Jersey 实现了 JSR-311 中对标注的支持，这使得开发人员使用 Java 开发 RESTfulWeb 服务更加容易。除此以外， Jersey 还增加了 JSR 以外的附加特性

Jersey

- 一个 feed 就是一个 POJO 类，至少要有有一个 method 有 @Path annotation
- 配置都使用 annotation，不再使用 XML

Jersey Demo

- @Path 的 hello World
-
- @Path 同时支持内嵌 param, 及 RE

Jersey Demo

- @Produces:(MIME)
- text/plain
- text/html
- application/xml
- application/json

Jersey Demo

- Extracting Path Parameters(@Path)
- Extracting Query Parameters(@QueryParam)
- Extracting Form Data
- Extracting the Java Type of a Req/Res

Jersey Demo

- @Products
- @Consumes
- MIME

Jersey Demo

- Injection(Support Custom Type)
- Property Injection
- Constructor Injection
- Sub-resource Injection