* 1. 同ASP.NET WebAPI中的依赖注入很类似
  + 相同点
    - 代码结构分为api层,service层和model层
    - 通过构造函数进行注入
    - 在统一的地方进行依赖注册
  + 不同点
    - 无接口的概念
* 2. 依赖注册主要分为以下几种类型
  + 注册一个值
* container.**registerValue**(name, [target, { **lifetime**: **Lifetime**.*SINGLETON* }]);
  + 注册一个函数（指定具体函数）
* **if** (target.**init**) {  
   container.**registerFunction**({ [name]: [target.**init**.bind(target),  
   {**lifetime**: **Lifetime**.*SINGLETON* }]});  
  }
  + 注册一个类
* *container.registerClass({[name]:[target.constructor,{lifetime:Lifetime.SINGLET}]})*
* 3.统一注册
  + 通过文件路径统一处理
* 注：
* 1. **大小写敏感**