链滴

CS 架构的 SDK 实现案例

作者：liumapp
原文链接：https://ld246.com/article/1530082559135
来源网站：链滴

<blockquote>
Server 端以 open-api 的形式对外开放服务，Client 端在集成 Server 端的 SDK 后，可以参照 SDK 的运行 Demo 来实现对 Server 的服务调用。
</blockquote>
<h2 id="前言">前言</h2>
先上案例源代码: Github: liumapp/simple-sdk-example
<h2 id="使用方法">使用方法</h2>

首先请确保您的操作系统包含了 Maven3、docker、docker-compose。
如果缺少的话，可以参考这个脚本进行安装：liumapp/install-docker-compose

安装镜像

执行 build-image.sh 脚本，完成 mysql、sdk-server、client-server 镜像的安装。

mysql 镜像用于支持 client-server 和 sdk-server 两个服务端
client-server 用于模拟 sdk-demo（sdk-core 是该服务的一个依赖）
sdk-server 用于模拟 sdk 的服务提供者。

启动

在项目根目录下，执行 docker-compose up -d，等待 10 秒左右后，在浏览器访问 localhost:2020

<h2 id="流程介绍">流程介绍</h2>
执行 docker-compose up -d 之后，等待 10 秒左右。
然后打开您的浏览器，分别访问

http://localhost:2020/
用于模拟客户端运行的 demo，依赖于 sdk-core，该服务会提供流程展示，具体流程为：

创建新的用户

提交用户的收货地址

用户下订单

获取订单详情

借助 sdk-core，向 sdk-server 调用订单数据备份接口，将订单详情数据备份到 sdk-server 的据库系统中。

借助 sdk-core，向 sdk-server 获取备份的订单数据，并打印在浏览器的 console 中。

http://localhost:2020/druid
方便查询 client-server 与数据库的交互。

http://localhost:3030/druid
方便查询 sdk-server 与数据库的交互。

<h2 id="开发调试">开发调试</h2>
<h3 id="后端">后端</h3>
因为系统是基于 docker 运行的，所以 client-server、sdk-server 和 mysql 之间都是以 docker-ompose 的 service name 来进行通讯的。
比如，在 client-server 下，它与 mysql 之间的连接建立地址为："jdbc:"
这里的 client-mysql 就是代表 docker-compose.yml 下 mysql 的 service name。
所以如果您要调试后端的话，并不建议直接修改 client-server 和 sdk-server 的配置文件。
我个人建议这种方式（假设您的操作系统为 Mac OS）：
修改您的/etc/hosts 文件，添加以下内容：
127.0.0.1 client-server
127.0.0.1 client-mysql
127.0.0.1 sdk-server
之后将项目导入您常用的 IDE 即可。
<h3 id="前端">前端</h3>
前端项目为 client-ui
项目的主要配置文件为./client-ui/static/js/config.js
前端项目可以独立在 nodejs 环境下运行，也可以做为 client-server 的静态资源去运行。
前者跟传统的前后端分离实现方式一样，后者需要额外对 client-ui 进行打包编译操作，然后运行 update-ui.sh 脚本将 dist 资源整合到 client-server 项目下。
<h2 id="注意事项">注意事项</h2>

需要确保您的系统具有 maven、docker、docker-compose 的支持，如果前端项目要独立运行话，还需要 nodejs 环境的支持。

如果要调试系统，请在您本地的 mysql 数据库执行 table.sql 来创建相关数据表。

