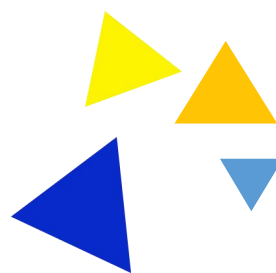


赛项样题

Web 技术



目录

一、介绍	4
二、项目和任务说明	4
第一阶段: RESTfull API	4
JSON 数据结构	5
错误处理	5
信息安全	7
用户登录	7
用户注册	7
商品分类列表 (所有)	11
商品分类列表 (分页)	11
删除商品分类	11
充值记录列表 (所有)	12
充值记录列表 (分页)	12
用户充值	12
创建商品	13
商品列表 (所有)	13
商品列表 (分页)	13
获取商品信息	13
修改商品信息	13
删除商品	14
创建商品订单	14
订单列表 (所有)	14
订单列表 (分页)	14
修改订单信息	15
获取订单信息	15
支付订单	15
地址列表 (所有)	15
地址列表 (分页)	16
新增地址	16

修改地址信息	16
删除地址	16
数据库	17
第二阶段：Vue 应用	18
Sunshop 页面	18
Sunshop Management 页面	19
三、模块区分	20
四、面向参赛者的提交说明	23

一、介绍

Sunshop 是中国本土一家专注数码产品销售的线上商城，他们目前需要对现有的线上商城系统进行重新开发，以适应于客户使用不同终端设备访问他们的商城进行浏览和选购等操作。

Sunshop 指定你依据 RESTful 规范完成 API 开发，并编写一个 Vue 应用完成线上商城功能，包括员工的商城管理页面，其中面向客户的商城页面（包括浏览、选购和订单等页面）需要支持不同类型的设备分辨率。

二、项目和任务说明

该项目分为两个阶段：

- 第一阶段：你需要在第一天（上午 2.5 小时，下午 2.5 小时）完成数据库与后端 RESTful API 的开发工作，包括员工的后台管理 API、客户的浏览选购 API。
- 第二阶段：你需要在第二天（上午 2.5 小时，下午 2.5 小时）完成 Vue 应用与页面的开发工作，包括员工的后台管理页面、客户的浏览选购等页面。你的 Vue 程序需要与你 Phase one 所完成的 RESTful API 进行对接，你可以对你第一天的 RESTful API 进行修改，但不会影响你 Phase one 的结果。

您所完成的项目需要分别部署到服务器中，评分时仅针对服务器内容进行判分：

- 第一阶段：将在第一天下午结束之后回收第一阶段作品，您 Phase one 完成的 API 应当通过 `http://<hostname>/API` 的 URL 进行访问，后续 API 开发均基于此 URL。
- 第二阶段：将在第二天下午结束之后回收第二阶段作品，您 Phase two 完成的 Vue 应用通过 `http://<hostname>/VUE` 的 URL 进行访问。
- 提示：URL 中的 `<hostname>` 将使用单独文件另行提供。

第一阶段：RESTful API

在此阶段，您需要根据提供的 RESTful API 的具体要求完成功能开发和数据库设计。所有 API 特定要求都有详细要求。

除本文档外，将会为您提供额外的文档用于说明 API 具体细节要求，这些文档包括：

- ◆ API 文档文件（html）
- ◆ Postman 文件（json）

JSON 数据结构

系统中 API 的返回数据应当遵守统一的 JSON 结构，所有数据需要以对象形式封装到结构中的 data 字段后返回。如以下示例所示：

JSON Data Structure
Status: 200 Content-Type: application/json Body: <pre>{ "status":200, "success":true, "message":"register successful", "data": { "nickname": "Jones" }, "time":"2021-08-23 14:50:28" }</pre> status - int, HTTP status code success - boolean, true when the API handles results, false on errors or failures message - string, message data - object null, the actual data, defined according to the API requirements time - datetime, response time

错误处理

在 API 中需要定义规范化的错误处理机制，如 404 错误、验证错误和认证错误等错误的提示数据也应当遵守前文的 JSON 数据结构的要求。如以下示例所示：

404 错误示例：当请求地址或资源不存在时

Not found
Status: 404

Content-Type: application/json

Body:

```
{
  "status": 404,
  "success": false,
  "message": "Not found",
  "time": "2022-05-23 14:41:53"
}
```

status - int, status code

success - boolean, This field must be false on API errors

message - string, error message

time - datetime, current time

验证错误示例：当请求数据不符合 API 接口的验证要求时

Validation error

Status: 422

Content-Type: application/json

Body:

```
{
  "status": 422,
  "success": false,
  "message": "Validation error",
  "time": "2022-05-23 14:41:53"
}
```

status - int, status code

success - boolean, This field must be false on API errors

message - string, error message

time - datetime, current time

认证错误示例：当请求需要认证的 API 时，未提供有效且正确的 token 时

Unauthorized

Status: 401

Content-Type: application/json

Body:

```
{
  "status": 401,
  "success": false,
  "message": "Unauthorized",
}
```

<pre>"time":"2022-05-23 14:41:53" }</pre>
status - int, status code
success - boolean, This field must be false on API errors
message - string, error message
time - datetime, current time

信息安全

API 需要进行一些安全限制，以防止 API 被黑客和恶意脚本滥用。您的安全措施至少包括以下形式：

- ◆ 对同一 IP 地址，同一 API 单位时间（60 秒）内的访问次数限制为 30 次。
 - 如果超过限制，应当返回一个符合<错误处理>段落要求的错误提示，其状态码应为 HTTP 429。

用户登录

请求用于用户登陆或身份验证。

如果客户端发送数据：

- ◆ 用户名及密码正确：HTTP 200
- ◆ 用户名或密码错误：HTTP 401，带有合理提示

在后续需要认证的 API 中，您需要在 HTTP header 中携带该 user token。

用户注册

请求用于注册一个新用户。

系统中应该有两种类型用户：

- ◆ user：普通用户，可以在商城中下单和结算
- ◆ staff：员工用户，可以进入管理后台进行管理

如果客户端发送数据：

- ◆ 注册成功：HTTP 200
- ◆ 数据有误，验证失败或登陆名重复：HTTP 422，带有合理提示

用户注销

请求登出当前用户。

如果客户端发送数据：

- ◆ 登出成功：HTTP 200
- ◆ user token 不合法：HTTP 401，带有合理提示

获取当前用户信息

请求用于获取当前用户信息。

如果客户端发送数据：

- ◆ 登出成功：HTTP 200
- ◆ user token 不合法：HTTP 401，带有合理提示

用户列表（所有）

[*Login required] [*Staff only] 请求用于获取全部用户列表。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200
- ◆ user token 不合法，权限不正确：HTTP 401，带有合理提示
- ◆ 请求数据错误：HTTP 422，带有合理提示

用户列表（分页）

[*Login required] [*Staff only] 请求用于获取一个用户列表。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200
- ◆ user token 不合法，权限不正确：HTTP 401，带有合理提示
- ◆ 请求数据错误：HTTP 422，带有合理提示

获取用户信息

[*Login required] [*Staff only] 请求用于获取一个用户。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200
- ◆ user token 不合法，权限不正确：HTTP 401，带有合理提示
- ◆ 未找到用户错误：HTTP 404，带有合理提示

删除用户

[*Login required] [*Staff only] 请求用于删除一个用户。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200
- ◆ user token 不合法，权限不正确：HTTP 401，带有合理提示
- ◆ 未找到用户错误：HTTP 404，带有合理提示

修改用户信息

[*Login required] [*Staff only] 请求用于修改一个用户昵称和权限。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200
- ◆ user token 不合法，权限不正确：HTTP 401，带有合理提示
- ◆ 未找到用户错误：HTTP 404，带有合理提示

上传图片

[*Login required] [*Staff only] 此请求可以将图片以 Base64 编码格式上传并返回图片信息。您需要传递一个对象，并在 header 中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，为上传图片信息
- ◆ 数据有误或验证失败：HTTP 422，返回验证错误，并应当有合理提示

删除图片

[*Login required] [*Staff only] 此请求可以删除一张图片，您必须在 URL 中传递删除的图片 id，并在 header 中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，为被删除的图片信息
- ◆ 未找到图片错误：HTTP 404，返回 404 错误

图片列表（所有）

[*Login required] [*Staff only] 此请求可以获取所有图片。您必须在 header

中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示全部数据
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误
- ◆ 请求数据错误：HTTP 422，返回验证错误

图片列表（分页）

[*Login required] [*Staff only] 此请求可以获取所有图片。您必须在 header 中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示全部数据
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误
- ◆ 请求数据错误：HTTP 422，返回验证错误

创建 Banner 图片

[*Login required] [*Staff only] 此请求可以创建一个 banner 图片。您必须传递一个对象，并在 header 中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示 banner 图片信息
- ◆ user token 不合法：HTTP 401，返回认证错误
- ◆ 数据有误或验证失败：HTTP 422，返回验证错误，并应当有合理提示

删除 Banner 图片

[*Login required] [*Staff only] 此请求可以删除 banner 图片。您必须在 URL 中传递删除的 Banner id，并在 header 中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示删除的 banner 图片信息
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误
- ◆ 数据有误或验证失败：HTTP 422，返回验证错误，并应当有合理提示
- ◆ 未找到 Banner 错误：HTTP 404，返回 404 错误

Banner 图片列表（所有）

此请求可以获取所有 banner 图片。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示全部数据

Banner 图片列表（分页）

此请求可以获取 banner 图片列表。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示全部数据
- ◆ 请求数据错误：HTTP 422，返回验证错误

创建商品分类

[*Login required] [*Staff only] 此请求可以创建一个商品分类。您必须在 header 中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回商品分类信息
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误
- ◆ 数据有误或验证失败：HTTP 422，返回验证错误，并应当有合理提示

商品分类列表（所有）

此请求可以获取所有商品分类。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回商品分类列表

商品分类列表（分页）

此请求可以获取所有商品分类。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回商品分类列表

删除商品分类

[*Login required] [*Staff only] 通过这个请求，可以删除商品分类。您必须

在 URL 中传递删除的分类 id，并在 header 中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回被删除的商品分类
- ◆ user token 不合法，权限错误：HTTP 401，返回认证错误
- ◆ 未找到商品分类错误：HTTP 404，返回 404 错误
- ◆ 被删除的商品分类下有子类或商品，禁止删除：HTTP 422，返回 422 错误

充值记录列表（所有）

[*Login required] [*Staff only] 此请求可以获取所有充值记录。您必须在 header 中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示全部数据
- ◆ user token 不合法：HTTP 401，返回认证错误

充值记录列表（分页）

[*Login required] [*Staff only] 此请求可以获取所有充值记录。您必须在 header 中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示全部数据
- ◆ user token 不合法：HTTP 401，返回认证错误

用户充值

[*Login required] [*Staff only] 此请求可以对用户充值。您必须传递一个对象，并在 header 中携带 user token。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，为充值用户信息
- ◆ user token 不合法，权限错误：HTTP 401，返回认证错误
- ◆ 数据有误或验证失败：HTTP 422，返回验证错误，并应当有合理提示

创建商品

[*Login required] [*Staff only] 此请求可以创建一个商品。您必须传递一个对象，并在 header 中携带 user token。

如果客户端发送数据：

- ◆ 创建成功：HTTP 200，返回内容为空
- ◆ user token 不合法：HTTP 401，返回认证错误
- ◆ 数据有误或验证失败：HTTP 422，返回验证错误，并应当有合理提示

商品列表（所有）

此请求可以获取所有商品。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示全部数据

商品列表（分页）

此请求可以获取所有商品。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示全部数据

获取商品信息

通过此请求，用户可以获取某个商品详细信息。您可以传递一个空对象，在 URL 中传递商品的 id。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，为商品详细信息
- ◆ 未找到商品错误：HTTP 404，返回 404 错误

修改商品信息

[*Login required] [*Staff only] 此请求可以修改商品信息。

如果客户端发送数据：

- ◆ 修改成功：HTTP 200，返回一个对象，为被修改的信息。
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误

- ◆ 未找到商品错误：HTTP 404，返回 404 错误

删除商品

[*Login required] [*Staff only] 此请求可以删除一个商品，必须使用软删除来进行删除操作。

如果客户端发送数据：

- ◆ 删除成功：HTTP 200，返回一个对象，为被删除的商品信息。
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误
- ◆ 未找到商品错误：HTTP 404，返回 404 错误

创建商品订单

[*Login required] 此请求可以创建商品订单，仅允许用户在前台浏览时进行下单操作。

如果客户端发送数据：

- ◆ 创建成功：HTTP 200，返回内容为空
- ◆ user token 不合法：HTTP 401，返回认证错误
- ◆ 数据有误或验证失败：HTTP 422，返回验证错误，并应当有合理提示

订单列表（所有）

[*Login required] 通过此请求，用户可以获取所有订单。

用户仅能读取自己的订单，而员工可以读取全部订单。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示所有数据

订单列表（分页）

[*Login required] [*Staff only] 通过此请求，用户可以获取订单列表。

用户仅能读取自己的订单，而员工可以读取全部订单。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示所有数据

修改订单信息

[*Login required] [*Staff only] 通过此请求，可以修改订单信息。

如果客户端发送数据：

- ◆ 修改成功：HTTP 200，返回一个对象，返回内容为空
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误
- ◆ 未找到订单错误：HTTP 404，返回 404 错误

获取订单信息

[*Login required] 通过此请求，可以获取订单信息，您可以传递一个空对象，在 URL 中传递订单的 id，并在 header 中携带 user token。用户仅能读取自己的订单，而员工可以读取全部订单。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，为一个订单信息。
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误
- ◆ 未找到订单错误：HTTP 404，返回 404 错误

支付订单

[*Login required] [*User only] 通过此请求，可以支付订单，您可以传递一个空对象，在 URL 中传递订单的 id。仅用户可以使用自己账号支付订单。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回内容为空，
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误
- ◆ 未找到订单错误：HTTP 404，返回 404 错误

地址列表（所有）

[*Login required] [*User only] 此请求可以获取所有地址，您必须在 header 中携带 user token。仅用户可以对自己的地址信息进行操作。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示全部数据

地址列表（分页）

[*Login required] [*User only] 此请求可以获取地址列表，您必须发送一个对象，并在 header 中携带 user token。仅用户可以对自己的地址信息进行操作。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，显示全部数据

新增地址

[*Login required] [*User only] 通过此请求，可以增加地址信息，您必须发送一个对象，并在 header 中携带 user token。仅用户可以对自己的地址信息进行操作。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，为被增加的地址信息。
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误

修改地址信息

[*Login required] [*User only] 此请求修改一个地址，您需要传递一个对象，在 URL 中传递修改的 address id，并在 header 中携带 user token。仅用户可以对自己的地址信息进行操作。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，为被修改的地址信息。
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误
- ◆ 未找到地址错误：HTTP 404，返回 404 错误

删除地址

[*Login required] [*User only] 此请求可以删除一个地址，您可以发送一个空对象，在 URL 中传递删除地址的 id，并在 header 中携带 user token。仅用户可以对自己的地址信息进行操作。

如果客户端发送数据：

- ◆ 操作成功：HTTP 200，返回一个对象，为被删除的地址信息。
- ◆ user token 不合法，权限不正确：HTTP 401，返回认证错误

◆ 未找到地址错误：HTTP 404，返回 404 错误

数据库

为您提供了一个现成的数据库，数据库中包含一些初始数据。您需要根据业务逻辑修改数据表结构或者创建新的数据表。

第二阶段：Vue 应用

在本阶段中，您需要根据提供的页面模版文件，使用 Vue 框架完成线上商场应用，您完成的应用需要对接你 Phase one 所完成的 API 接口，并遵循 Phase one 中规定的业务逻辑。

你可以对你的 API 进行一些必要的修改，以便于你完成本阶段的开发。在本阶段你的任务被分为两类，分别为面向客户的商城页面和面向员工的商城管理页面。

您完成的 Vue 应用功能需要符合以下原则：

- ◆ 功能逻辑遵守<第一阶段>中规定的业务逻辑。
- ◆ 功能逻辑与数据与提供的页面模版匹配。
- ◆ 针对用户的误操作和 API 报错有良好的错误提示与交互效果，任何错误不会导致单页应用无法正常运行。

Sunshop 页面

商城页面主要面向普通客户的浏览、选购等功能，客户可能通过不同的终端设备访问在线商城，您需要对提供的页面模版进行一些响应式优化，至少支持两种设备分辨率：1920px（PC），480px（Mobile）。

对于没有提供模版的页面，您需要自行完成页面布局，布局风格需要与其他模版页面一致。

页面和功能包括但不限于：

- 首页
 - 商品的分类菜单
 - 能够跳转到用户中心页面、购物车页面
 - 具备轮播功能的 Banner 图片
 - 热门分类及其主要商品列表
 - 热门商品列表
 - Management Page 入口，仅限员工账号时可用
- 购物车页面
 - 增加、删除和修改购物车内商品
 - 选择商品跳转至下单页面【员工不可用】
- 下单页面【员工不可用】

- 选择和编辑收货地址
- 列出订单内商品信息、订单价格等信息
- 提交订单后跳转支付页面
- 支付页面【员工不可用】
 - 显示提交的订单信息
 - 对订单进行确认支付
- 商品列表页
- 商品详情页
 - 显示商品详细信息
- 用户中心页【员工不可用】
 - 用户信息 – 包括用户名、余额等
 - 进入我的收藏、订单列表、充值记录、地址管理等页面
- 我的收藏页【员工不可用】
 - 列出收藏
 - 取消收藏
- 订单列表页【员工不可用】
 - 列出订单与订单基础信息
 - 进入订单详情页
- 订单详情页【员工不可用】
 - 列出订单详细信息
 - 下单未支付订单可以跳转支付页面重新支付
- 充值记录页【员工不可用】
 - 列出充值记录
- 地址管理页【员工不可用】
 - 列出地址信息
 - 增加、删除和修改地址信息

Sunshop Management 页面

商城管理页面用于员工对商城进行各类管理功能。你需要根据提供的模版页面完成有关商城管理功能的开发，而对于没有提供模版的页面，您需要自行完成页面布局，布局风格需要与其他模版页面一致。

页面和功能包括但不限于：

- 用户管理
 - 列表页面 - 包括删除、修改、充值等入口
 - 修改页面 - 可以为弹出的模态框
 - 充值页面 - 可以为弹出的模态框
- 分类管理
 - 列表页面 - 包括删除、修改、创建等入口
 - 修改页面 - 可以为弹出的模态框
 - 创建页面 - 可以为弹出的模态框
- 商品管理
 - 列表页面 - 包括删除、修改、创建等入口
 - 修改页面 - 可以为弹出的模态框
 - 创建页面 - 可以为弹出的模态框
- 订单管理
 - 列表页面 - 包括修改状态、修改金额等入口
 - 修改状态页面 - 可以为弹出的模态框
 - 修改金额页面 - 可以为弹出的模态框
- 充值管理
 - 列表页面
 - 充值页面 - 可以为弹出的模态框

三、模块区分

竞赛时会按照模块区分不同阶段的内容，您的作品应在模块规定的时间内完成：

模块 A （Sunshop 页面 API 开发）

- 用户
 - ◆ 用户登录
 - ◆ 用户注册
 - ◆ 用户注销
 - ◆ 获取当前用户信息
- 商品分类
 - ◆ 商品分类列表（所有）

- ◆ 商品分类列表（分页）
- 充值
 - ◆ 充值记录列表（所有）
 - ◆ 充值记录列表（分页）
- 商品
 - ◆ 商品列表（所有）
 - ◆ 商品列表（所有）
 - ◆ 获取商品信息
- 订单
 - ◆ 创建商品订单
 - ◆ 订单列表（所有）
 - ◆ 订单列表（分页）
 - ◆ 获取订单信息
 - ◆ 支付订单
- 地址
 - ◆ 地址列表（所有）
 - ◆ 地址列表（分页）
 - ◆ 新增地址
 - ◆ 修改地址信息
 - ◆ 删除地址

模块 B（Sunshop Management 页面 API 开发）

- 用户
 - ◆ 用户列表（所有）
 - ◆ 用户列表（分页）
 - ◆ 修改用户信息
 - ◆ 删除用户
 - ◆ 获取用户信息
- 图片
 - ◆ 上传图片
 - ◆ 删除图片
 - ◆ 图片列表（所有）

- ◆ 图片列表（分页）
- Banner 图片
 - ◆ 创建 Banner 图片
 - ◆ 删除 Banner 图片
 - ◆ Banner 图片列表（所有）
 - ◆ Banner 图片列表（分页）
- 商品分类
 - ◆ 创建商品分类
 - ◆ 删除商品分类
- 充值
 - ◆ 用户充值
- 商品
 - ◆ 创建商品
 - ◆ 修改商品信息
 - ◆ 删除商品
- 订单
 - ◆ 修改订单信息

模块 C （Sunshop 页面开发）

- Sunshop 页面
 - ◆ 首页
 - ◆ 购物车页面
 - ◆ 下单页面
 - ◆ 支付页面
 - ◆ 商品列表页
 - ◆ 商品详情页
 - ◆ 用户中心页
 - ◆ 我的收藏页
 - ◆ 订单列表页
 - ◆ 订单详情页
 - ◆ 充值记录页
 - ◆ 地址管理页

模块 D （Sunshop Management 页面开发）

- Sunshop Management 页面

- ◆ 用户管理
- ◆ 分类管理
- ◆ 商品管理
- ◆ 订单管理
- ◆ 充值管理

注意：模块 A 中部分 API 接口可能与 Sunshop Management 页面有关！

四、面向参赛者的提交说明

评分时仅针对服务器内容进行判分，您的作品应按照以下路径部署到服务器中：

- 第一阶段 - RESTful API

- 在您部署的应用以及数据库中，您的应用初始管理员账号及密码必须被定义，并保证可用。

- ◆ UserName: admin

- ◆ Password: sunshop

- API 应该部署到 URL: `http://<hostname>/API`

- 您设计的数据库结构和数据（包含完整可运行的数据）应以 `db-dump.sql` 文件名导出到 URL: `http://<hostname>/db-dump`

- 第二阶段 - Vue Application

- Vue 应用应该部署到 URL: `http://<hostname>/VUE`

- Note: URL 中的 `<hostname>` 将使用单独文件另行提供。