

作业-2

规范

提交

检查提交

收集提交内容

适用于演员/女演员的 REST API

在此任务中，您需要开发一个 Flask-Restx 数据服务，该服务允许客户端读取和存储有关演员/女演员的信息，并允许使用者通过 REST API 访问数据。

该任务基于电视迷宫API，该API提供了电视节目和人物的详细列表。您可以使用以下链接浏览 TVmaze API

- 源网址：(<http://api.worldbank.org/v2/>) <http://api.tvmaze.com/>
(<http://api.tvmaze.com/>)
- (<http://api.worldbank.org/v2/>) 有关 API 调用结构的文档：
<https://www.tvmaze.com/api> (<https://www.tvmaze.com/api>)

免责声明：我们正在使用TV Maze (<https://www.tvmaze.com/>) 提供的极值API。我们希望学生与现实生活中的网络服务进行交互，以增加学习体验，因此我们对API中实体和描述的措辞或包含/排除概不负责。这是一个"构建 REST API"练习，应该这样对待。

在此分配中，您将使用此 API 中提供的信息，并添加下面列出的一些功能：

分配规范

问题1：添加新的演员（3分）

此操作可被视为按需"导入"操作，以获取执行组件的详细信息以将其存储在应用程序中。该服务将下载给定演员/女演员（按其名称）的JSON数据；导入执行组件后，必须使用 **SQLite** 在本地存储数据（数据库的名称应为 **YOURZID.db**）。

您可以使用以下内容来查询 API： [https://api.tvmaze.com/search/people?q=?](https://api.tvmaze.com/search/people?q=)
([https://api.tvmaze.com/search/people?q=?](https://api.tvmaze.com/search/people?q=))

例如，您可以检查以下查询： <https://api.tvmaze.com/search/people?q=Brad%20Pitt>
(<https://api.tvmaze.com/search/people?q=Brad%20Pitt>)

要添加执行组件，您的 API 会接受一个名为"name"的查询参数：

- **姓名**： 演员姓名

导入集合后，服务应返回至少包含以下信息的响应：

- **id**： 自动生成的唯一整数标识符（这可能与 tvmaze API 中的 actor ID 不同）
- **上次更新时间**： 集合存储在数据库中的时间
- **_links**： 可以使用其检索导入集合的 URL（如下面的示例所示）

重要事项： 对于此 分配，系统会要求您以编程方式访问给定的 Web 内容。某些 Web 主机不允许以编程方式访问其网页，如果您访问其网页的次数过多，某些主机可能会阻止您的 IP。在实现过程中，下载一些测试用例并在本地访问内容 - 尽量不要以编程方式执行太多 HTTP 请求。查看 tvmaze API 的文档，以获取有关其速率限制策略的见解。

例：

```
POST /actors?name=brad pitt
```

示例响应 [这不是您在 DB 中存储的内容，而是呼叫的响应]

： 201 已创建

```
{
  "id" : 123,
  "last-update": "2021-04-08-12:34:40",
  "_links": {
    "self": {
      "href": "http://[HOST_NAME]:[PORT]/actors/123"
    }
  }
}
```

- 如果只有名字完全匹配（"brad pitt", "brad-pitt", "Brad pitt", "Brad pitt"彼此匹配，但它们不应匹配像"bard pit"这样的名字），则必须导入演员（例如Brad Pitt）。请注意，TVMaze API提供了模糊搜索，因此可以容忍拼写错误，大写/小号，破折号...等，同时提供比确切名称更多的结果。您只应导入匹配的字符（忽略大小写以及除英文字母、**空格**和数字之外的任何字符）。如果有几个演员的名字相同，你只需要导入其中一个（任何）

提示：在将名字发送到tvmaze API之前，将所有非英语字符（例如，，-，_，?）替换为空格，然后将其发送到API；因此，"brad pit-t"将转换为"brad pit t"，这将与"brad pitt"不匹配。点不会使这变得非常复杂；我们不希望您处理所有异常等。

Keep 就像规范中提到的一样简单。

- 您在数据库中存储数据的内容和方式取决于您，但请查看其余问题，以了解您需要为每个Actor保留哪些属性。至少你需要存储："姓名"，"国家"，"生日"，"死亡日"，"性别"以及他/她的"节目"列表。导入演员时，还应存储他/她的所有"节目"。为此，您需要探索"TVmaze"API，看看如何自动获取这样的列表。

- 切勿更改资源的 ID（不要更新 ID）
- 应将 [HOST_NAME]: [PORT] 替换为正确的值
- 关于 _links 的简要说明如下： <https://developer.wordpress.org/rest-api/using-the-rest-api/linking-and-embedding> (<https://developer.wordpress.org/rest-api/using-the-rest-api/linking-and-embedding/>) /

问题 2 - 检索参与者（1.5 分）

此操作 通过执行组件的 ID（由应用程序生成的 ID）检索执行组件。此操作的响应将显示执行组件的详细信息。请参阅下面提供的响应示例，了解响应中应包含哪些属性。"_links"提供上一个、下一个和当前资源（如果存在）的链接。下一个和上一个资源基于应用程序生成的顺序 ID。

界面应如下所示：

```
GET /actors/{id}
```

示例响应返回：200 正常

```
{
  "id": 124,
  "last-update": "2022-03-08-12:34:40",
  "name": "Some One",
  "country": "Australia",
  "birthday": "22-05-1987",
  "deathday": null,
  "shows": [
    "show 1",
    "show 2",
    "show 3"
  ],
  "_links": {
    "self": {
      "href": "http://[HOST_NAME]:[PORT]/actors/124"
    },
    "previous": {
      "href": "http://[HOST_NAME]:[PORT]/actors/123"
    },
    "next": {
      "href": "http://[HOST_NAME]:[PORT]/actors/125"
    }
  }
}
```

问题3-删除演员（1.5分）

此操作将从数据库中删除现有执行组件。界面应如下所示：

```
DELETE /actors/{id}
```

返回：200 确定

```
{
  "message" : "The actor with id 134 was removed from the database!",
  "id": 134
}
```

问题 4 - 更新演员（2 分）

此操作将部分更新给定执行组件的详细信息。

界面应类似于以下示例：

```
PATCH /actors/{id}
{
  "name": "Some One",
  "country": "Australia",
  "birthday": "22-05-1987",
  "deathday": null
}
```

上面的有效载荷只是一个例子;它可以包含任何演员的属性。请查看问题 2 中的示例响应以了解现有属性。

返回：200 确定

```
{
  "id" : 123,
  "last-update": "2021-04-08-12:34:50",
  "_links": {
    "self": {
      "href": "http://[HOST_NAME]:[PORT]/actors/123"
    }
  }
}
```

问题 5 - 检索可用演员列表 (4 分)

此操作将检索所有可用的参与者。接口应为：

```
GET /actors?order=<CSV-FORMATED-VALUE> & page=1 & size=10 & filter=<CSV-FORMAT
```

所有四个参数都是可选的，默认值为"order=+id", "page=1"和"size =10", filter="id, name"。 "页面"和"大小"用于分页;"size"显示每页的参与者数量。"order"是一个逗号分隔的字符串值，用于根据给定的条件对列表进行排序。该字符串由两部分组成：第一部分是特殊字符"+"或"-", 其中"+"表示升序, "-"表示降序。第二部分是属性名称，它是{id, 名称, 国家/地区, 生日, 死亡日, 上次更新}之一。以下是"订单"的一些示例值：

+姓名, +身份证	按"名称升序"和"id升序" 在本例中，按"名称"排序的 sorti ng 比 "id" 具有 p 的相关性。这类似于 SQL 按子句排序： " 名称 ASC, id ASC "
- 国家	按"国家/地区降序"排序

"filter"也是另一个逗号分隔的值 (id, 姓名, 国家/地区, 生日, 死亡日, 上次更新, 节目) , 并相应地显示每个演员应显示的属性。请看以下示例：

```
GET /actors?order=+id&page=1&size=10&filter=id,name
```

所有四个参数都是可选的，默认值为"order=+id"、"page=1"和"size=10"、"filter=id, name"
返回：200 确定

```
{
  "page": 1,
  "page-size": 10,
  "actors": [
    {
      "id" : 1,
      "name" : "Brad Pitt"
    },
    {
      "id" : 2,
      "name" : "Angelina Jolie"
    },
    ...
  ],
  "_links": {
    "self": {
      "href": "http://[HOST_NAME]:[PORT]/actors?order=+id&page=1&size=10&f"
    },
    "next": {
      "href": "http://[HOST_NAME]:[PORT]/actors?order=+id&page=2&size=10&f"
    }
  }
}
```

问题6 - 获取现有Actor的统计数据（3分）

此操作接受名为"format"的参数，该参数可以是"json"或"image"。根据格式，您的操作应提供以下信息：如果格式为图像，则操作应返回图像（可以是任何图像格式），并且图像在可视化中说明请求的信息（在创建可视化时应用您的所有知识，例如选择适当的可视化类型并确保它是人类可读的，清晰，信息丰富）。

- Actor按由"by"参数（逗号分隔的值）确定的属性进行细分;此参数可以是以下任何参与者属性："国家/地区"（显示每个国家/地区的参与者百分比）、"生日"、"性别"和"life_status"（表示活着的参与者的百分比）。您可以根据给定的属性决定使用不同类型的图表。
- 参与者总数
- 过去 24 小时内更新的执行组件总数
- 更新：这两个参数都是必需的

当格式为 JSON 时，界面应如下所示：

```
GET /actors/statistics?format=json&by=country,gender
```

返回: 200 确定

```
{
  "total": 1241,
  "total-updated": 24,
  "by-country" : { "Australia": 60.7, "USA": 19.2, ... },
  "by-gender" : { "Male": 48.6, "Femail": 51.4}
}
```

通知:

- 只能使用实验室中使用的库（最新版本）
- TVMAZE API 仅在问题 1 中用于导入新的参与者。其余操作依赖于本地数据库中的现有参与者
- 此作业没有模板代码，您的提交应坚持作业规范。
- 您的提交将被手动标记。
- 您应该遵守讲座中提到的REST API的最佳设计指南（例如，基于JSON格式的适当响应和正确的状态代码，完整的API文档：生成的Swagger应该完全不言自明，包括操作摘要，参数描述，默认值）。
- 应考虑标题无效或任何无效尝试使用终结点等情况（例如，如果数据源中不存在输入标题，则返回错误 404）
- 如果已导入集合，则应返回适当的响应（JSON）
- 您的代码必须在 **flask-restx** 中实现，并自动生成用于测试端点的 swagger 文档。
- 您的代码必须在 CSE 计算机中可执行
- 您的代码不得要求安装任何软件（python库很好）
- 您的代码必须用 Python 3.5 或更高版本编写。
- 您的操作（API 方法）必须以 JSON 格式返回适当的响应，以及相应的 HTTP 响应代码！例如，**500内部服务器错误** 是不适当的响应！
- 确保在数据库和 API 方法中使用了正确的数据类型（例如，不是年份为"2017"的字符串）
- 某些操作的一些响应包含"_links"，根据响应，这应该包括指向"self"，"next"和"previous"资源的链接。

常见问题:

- **日期-时间格式是什么？**
您可以使用任何格式（人类可读），但对于所有日期/时间值（包括生日，上次更新）都应该如此
- **如何获取演员的 SHOW？**
这是任务的一部分，您应该探索 API 并找到一种链接演员和节目的方法
- **我可以使用的哪些库？**
您可以使用在任何实验室中导入的所有库，以及 SQLAlchemy

- **API 用户能否更新 ID、链接和上次更新时间？**

不可以，用户无法编辑这些属性。不应更改 ID，链接和上次更新字段不是执行组件属性的一部分。

提交：

- 截止日期为**2022年4月4日星期一20:00**

- 一个且唯一——一个名为"YOUR_ZID.py"的Python脚本文件，其中包含您的代码

- **如何提交作业？**

转到作业页面，单击"提交"选项卡;选择必须命名为"YOUR_ZID.py"的文件。确保文件不为空，然后一起提交文件。

- **我可以在截止日期之后提交我的文件吗？**

是的，可以。但是，您作业的5%将作为每天的滞纳金扣除。5天后不接受延迟提交。

剽窃

这是一个单独的分配。您提交的作品必须是您自己的作品。不允许提交部分或完全来自任何其他人与任何其他共同撰写的作品。对这种罪行的惩罚可能包括负面分数，课程的自动失败，以及可能的另一门学科。作业提交将手动审核。

不要向任何其他人提供或展示您的作业 - 除了本课程的教学人员。如果您出于任何原因故意向另一个人提供或展示您的作业作品，并且提交了由此产生的作品，则即使该作品是在您不知情或未经您同意的情况下提交的，您也可能受到处罚。请注意，**保护您的代码工件也是您的职责**。如果您使用在线解决方案来存储代码工件（例如，GitHub），请确保将存储库保密，并且不要与任何人共享访问权限。

提醒：剽窃被定义为 (<https://student.unsw.edu.au/plagiarism>)使用他人的话语或想法，并将它们呈现为您自己的。新南威尔士大学和CSE将剽窃视为学术不端行为，这意味着它受到的处罚与被排除在新南威尔士大学进一步研究之外一样严重。有几个在线资源可以帮助您了解什么是抄袭以及新南威尔士大学如何处理抄袭：

- 抄袭和学术诚信 (<https://student.unsw.edu.au/plagiarism>)
- 新南威尔士大学抄袭程序 (<https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf>)

确保您阅读并理解此内容。无知不被接受为剽窃的借口。特别是，您还负责确保除了您之外的任何人都无法访问您的分配文件，方法是在CSE目录和代码存储库中设置正确的权限（如果使用）（例如，Github和类似权限）。还要注意，抄袭包括付钱或要求别人为您做一件作品，然后将其作为你自己的作品提交。

新南威尔士大学一直致力于培养一种以学术诚信为依据的学习文化。所有新南威尔士大学教职员工和学生都有责任坚持这一学术诚信原则。剽窃会破坏学术诚信，在新南威尔士大学是不能容忍的。

资源创建于 [24 days ago \(Friday 11 March 2022, 03:20:24 PM\)](#)，上次修改时间为 [6 days ago \(Wednesday 30 March 2022, 07:37:04 AM\)](#)。