

Problem Statement

Create a very simple Django/flask project for managing the expenses of a company. The company has several employees and vendors. Each employee can make a number of expenses to each vendor. Create a system to add employees, add vendors and expenses via Rest APIs. Also, create APIs to fetch all expenses done by the employees and all expenses which are done to a vendor.

You need to create the following APIs as per the specification below. A script has been provided to check if the APIs are matching to all specifications are not. The script contains the exact API specification and does scoring of the APIs you are going to create. Your project should try to get a full score (100).

Note the dependency for the script: **requests**, you can install it by running **pip install requests**

Script usage: Run it from the command: **"python test.py <URL where your django project is running>"** The script will output your score along with details on each request it is sending to your server, actual response & response code received and expected response & response code.

Note on the URL to be passed:

If in postman, you are hitting the add-employee URL at <http://127.0.0.1:8000/add-employee/>, then pass the second argument as <http://127.0.0.1:8000/>

Similarly, if you are hitting it at <http://localhost:5000/expense/add-employee/> in postman, then pass the second argument as <http://localhost:5000/expense/>

The test script will print the actual request URL. Use that to check if you are passing the correct URL or have created correct URL or not.

A brief description of APIs is below:

API 1: Create Employee API:

URL: /add-employee/

HTTP Method: POST

Parameters:

name: Name of the employee.

employee_code: Employee code - this should be unique.

Success Response: {"message": "employee created."}

API 2: Get Employee API:

URL: /get-employee/?employee_code=<employee code>

HTTP Method: GET

Parameters:

employee_code: Employee code of the employee code you want.

Success Response: {"name": "Employee name", "employee_code": "EMP-001"}

API 3: Create Vendor API:

URL: /add-vendor/

HTTP Method: POST

Parameters:

name: Name of the vendor.

vendor_code: Vendor code - this should be unique.

Success Response: {"message": "vendor created."}

API 4: Get Vendor API:

URL: /get-vendor/?vendor_code=<vendor code>

HTTP Method: GET

Parameters:

vendor_code: Vendor code of the vendor you want detail of.

Success Response: {"name": "Vendor name", "vendor_code": "VND-001"}

API 5: Add Expense API:

URL: /add-expense/

HTTP Method: POST

Parameters:

vendor_code: Vendor code of the expense.

employee_code: Employee code of the expense.

expense_comment: Comment on the expense.

expense_done_on: date on which expense was done, in the format: "DD-MMM-YYYY", e.g. for 12th Aug 2019, it would be passed as "12-Aug-2019"

expense_amount: integer amount of expense.

Success Response: {"message": "expense created."}

API 6: Get Employee Expense API:

URL: /get-expense-for-employee/?employee_code=<employee code>

HTTP Method: GET

Return all expenses for the employee code.

Success Response: {"name": <employee name>, "expenses": [{"vendor": <vendor name>, "expense_made_on": <expense date>, "expense_comment": <expense comment>, "expense_amount": <expense amount>,}]}

API 7: Get Vendor Expense API:

URL: /get-expense-for-vendor/?vendor_code=<vendor code>

HTTP Method: GET

Return all expenses for the vendor code.

Success Response: {"name": <vendor name>, "expenses": [{"employee": <employee name>, "expense_made_on": <expense date>, "expense_comment": <expense comment>, "expense_amount": <expense amount>,}]}