




Django Restful API 1.0.0 OAS 3.0

/expense/api/schema/

Django Restful API. It is developed using DjangoRestFramework, ORM, MySQL, SQLite3, Swagger for api documentation. Main Features all kind api enpoints to perform CRUD using API for example We have endpoints for Add New Expense, Get All Expenses, Get Single Expense ,Udpate, Delete. All api endpoints are wel tested via performing unit test and documented using swagger to enable other developer to use it.

Authorize 

expense

GET	/expense/api/expenses/	GetAll Data		
POST	/expense/api/expenses/	Add new data item		
GET	/expense/api/expenses/{expId}/	Get single data item		
PUT	/expense/api/expenses/{expId}/	Update data completely		
PATCH	/expense/api/expenses/{expId}/	Update data item partially		
DELETE	/expense/api/expenses/{expId}/	Delete expense item		
GET	/expense/api/expenses/filter/by/{searchedKey}/			

Authorize



expense



GET

/expense/api/expenses/ Get All Data



POST

/expense/api/expenses/ Add new data item



GET

/expense/api/expenses/{expId}/ Get single data item



PUT

/expense/api/expenses/{expId}/ Update data completely



PATCH

/expense/api/expenses/{expId}/ Update data item partially



DELETE

/expense/api/expenses/{expId}/ Delete expense item



GET

/expense/api/expenses/filter/by/{searchedKey}/



Schemas



Expense >

No parameters

Responses

Code	Description	Links
200		No links

Media type

application/json

Examples

Valid Response 1

Controls Accept header.

Example Value | Schema

```
{
  "all_expenses": [
    {
      "pk": 1,
      "exp_user": "krishna",
      "exp_date": "2021-02-12",
      "exp_amount": 4100,
      "exp_title": "Car Services Repair",
      "exp_description": "Sample of description: An automobile repair shop (also known regionally as a garage or a workshop) is an establishment "
    },
    {
      "pk": 2,
      "exp_user": "admin",
      "exp_date": "2021-02-12",
      "exp_amount": 2100,
      "exp_title": "Mobile Services Repair",
      "exp_description": "Sample of description: An automobile repair shop (also known regionally as a garage or a workshop) is an establishment "
    },
    {
      "pk": 3,
      "exp_user": "krish",
      "exp_date": "2023-02-12",
      "exp_amount": 4004,
      "exp_title": "LPU Collage Fees",
      "exp_description": "Sample of description: An automobile repair shop (also known regionally as a garage or a workshop) is an establishment "
    }
  ]
}
```

GET

/expense/api/expenses/

Get All Expenses

This endpoint will return all the expense items from database

Parameters

No parameters

Try it out

Responses

Code	Description	Links
200	<div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div>Example Value Schema</div><div><pre>{ "all_expenses": [{ "pk": 1, "exp_user": "krishna", "exp_date": "2021-02-12", "exp_amount": 4100, "exp_title": "Car Services Repair", "exp_description": "Sample of description: An automobile repair shop (also known regionally as a garage or a workshop) is an establishment " }, { "pk": 2, "exp_user": "admin", "exp_date": "2021-02-12", "exp_amount": 2100, "exp_title": "Mobile Services Repair", "exp_description": "Sample of description: An automobile repair shop (also known regionally as a garage or a workshop) is an establishment " }, { "pk": 3, "exp_user": "krish", "exp_date": "2023-02-12", "exp_amount": 4004, "exp_title": "LPU Collage Fees", "exp_description": "Sample of description: An automobile repair shop (also known regionally as a garage or a workshop) is an establishment " }]}</pre></div></div> <div>Examples</div> <div>Valid Response 1</div>	

No links

 |

POST

/expense/api/expenses/

Add new expense item

This endpoints are use add new expense item in our database using

Parameters

Try it out

No parameters

Request body

application/json

Example Value

Schema

```
{
  "exp_date": "2023-08-10",
  "exp_title": "string",
  "exp_user": "string",
  "exp_amount": 0,
  "exp_description": "string"
}
```

Responses

Code	Description	Links
201	<div><div>Media type</div><div>application/json</div><div>Controls Accept header</div><div>Example Value</div><div>Schema</div></div> <div><div>Examples</div><div>Valid Response 1</div></div> <div><pre>{ "data": [{ "msg": "Congrats! expense successfully inserted", "data": { "pk": 1, "exp_user": "Irishna", "exp_date": "2021-02-12", "exp_amount": 4100, "exp_title": "Car Services Repair", "exp_description": "Sample of description: An automobile repair shop (also known regionally as a garage or a workshop) is an establishment " } }] }</pre></div>	

 No links |

Responses

Code	Description	Links
200		No links

Media type

application/json

Examples

Valid Response 1

Controls Accept header.

Example Value | Schema

```
{
  "data": [
    {
      "msg": "Detail of expense of id: 1",
      "data": {
        "pk": 1,
        "exp_user": "krishna",
        "exp_date": "2021-02-12",
        "exp_amount": 4100,
        "exp_title": "Car Services Repair",
        "exp_description": "Sample of description: An automobile repair shop (also known regionally as a garage or a workshop) is an establishment "
      }
    }
  ]
}
```

PUT /expense/api/expenses/{expId}/ Update expense completely

PATCH /expense/api/expenses/{expId}/ Update expense partially

DELETE /expense/api/expenses/{expId}/ Delete expense item

Schemas

```
{
  "exp_date": "2023-08-10",
  "exp_title": "string",
  "exp_user": "string",
  "exp_amount": 0,
  "exp_description": "string"
}
```

Responses

Code	Description	Links
200		No links

Media type

application/json

Examples

Valid Response 1

Controls Accept header.

Example Value | Schema

```
{
  "data": [
    {
      "msg": "Congrats expense successfully updated",
      "data": {
        "pk": 1,
        "exp_user": "krishna",
        "exp_date": "2021-02-12",
        "exp_amount": 4100,
        "exp_title": "Car Services Repair",
        "exp_description": "Sample of description: An automobile repair shop (also known regionally as a garage or a workshop) is an establishment "
      }
    }
  ]
}
```


Responses

Code	Description	Links
201	<div><div>Media type</div><div>application/json</div></div> <div>Examples</div> <div>Valid Response 1</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <pre>{ "data": [{ "msg": "Congrats! expense successfully inserted", "data": { "pk": 1, "exp_user": "krishna", "exp_date": "2021-02-12", "exp_amount": 4100, "exp_title": "Car Services Repair", "exp_description": "Sample of description: An automobile repair shop (also known regionally as a garage or a workshop) is an establishment " } }] }</pre>	No links

GET	/expense/api/expenses/{expId}/	Get single expense item	✓	🔒
PUT	/expense/api/expenses/{expId}/	Update expense completely	✓	🔒
PATCH	/expense/api/expenses/{expId}/	Update expense partially	✓	🔒
DELETE	/expense/api/expenses/{expId}/	Delete expense item	✓	🔒

Authorize

expense

GET /expense/api/expenses/ Get All Data

POST /expense/api/expenses/ Add new data item

GET /expense/api/expenses/{expId}/ Get single data item

PUT /expense/api/expenses/{expId}/ Update data completely

PATCH /expense/api/expenses/{expId}/ Update data item partially

DELETE /expense/api/expenses/{expId}/ Delete expense item

GET /expense/api/expenses/filter/by/{searchedKey}/

Schemas

Expense >

Django Restful API 1.0.0 OAS 3.0

/expense/api/schema/

Django Restful API. It is developed using DjangoRestFramework, ORM, MySQL, SQLite3, Swagger for api documentation. Main Features all kind api endpoints to perform CRUD using API for example We have endpoints for Add New Expense, Get All Expenses, Get Single Expense ,Udpate, Delete. All api endpoints are wel tested via performing unit test and documented using swagger to enable other developer to use it.

Authorize

expense

GET /expense/api/expenses/ Get All Data

POST /expense/api/expenses/ Add new data item

GET /expense/api/expenses/{expId}/ Get single data item

PUT /expense/api/expenses/{expId}/ Update data completely

PATCH /expense/api/expenses/{expId}/ Update data item partially

DELETE /expense/api/expenses/{expId}/ Delete expense item

GET /expense/api/expenses/filter/by/{searchedKey}/