**FastAPI Multiple Routing Example**

Here’s an example of a \*\*FastAPI\*\* application with multiple routes organized into different files for better modularity and maintainability. This example demonstrates how to structure a FastAPI project with multiple routing modules.  
  
---  
  
### Project Structure  
```  
fastapi\_app/  
├── main.py  
├── routers/  
│ ├── users.py  
│ ├── items.py  
│ └── \_\_init\_\_.py  
```  
  
---  
  
### Code Example  
  
#### 1. `main.py`  
This is the entry point of the application. It imports and includes the routers defined in the `routers` directory.  
  
```python  
from fastapi import FastAPI  
from routers import users, items  
  
app = FastAPI()  
  
# Include routers  
app.include\_router(users.router, prefix="/users", tags=["Users"])  
app.include\_router(items.router, prefix="/items", tags=["Items"])  
  
@app.get("/")  
def read\_root():  
 return {"message": "Welcome to the FastAPI application!"}  
```  
  
---  
  
#### 2. `routers/users.py`  
This file defines routes related to user operations.  
  
```python  
from fastapi import APIRouter  
  
router = APIRouter()  
  
# Example route to get all users  
@router.get("/")  
def get\_users():  
 return {"users": ["Alice", "Bob", "Charlie"]}  
  
# Example route to get a specific user by ID  
@router.get("/{user\_id}")  
def get\_user(user\_id: int):  
 return {"user\_id": user\_id, "name": f"User {user\_id}"}  
  
# Example route to create a new user  
@router.post("/")  
def create\_user(name: str):  
 return {"message": f"User {name} created successfully!"}  
```  
  
---  
  
#### 3. `routers/items.py`  
This file defines routes related to item operations.  
  
```python  
from fastapi import APIRouter  
  
router = APIRouter()  
  
# Example route to get all items  
@router.get("/")  
def get\_items():  
 return {"items": ["Item1", "Item2", "Item3"]}  
  
# Example route to get a specific item by ID  
@router.get("/{item\_id}")  
def get\_item(item\_id: int):  
 return {"item\_id": item\_id, "name": f"Item {item\_id}"}  
  
# Example route to create a new item  
@router.post("/")  
def create\_item(name: str):  
 return {"message": f"Item {name} created successfully!"}  
```  
  
---  
  
#### 4. `routers/\_\_init\_\_.py`  
This file is required to make the `routers` directory a Python package. It can be left empty or used for package-level imports.  
  
```python  
# This file can be empty or used for package-level imports  
```  
  
---  
  
### How to Run the Application  
1. Install FastAPI and Uvicorn if you haven’t already:  
 ```bash  
 pip install fastapi uvicorn  
 ```  
  
2. Run the application:  
 ```bash  
 uvicorn main:app --reload  
 ```  
  
3. Open your browser and navigate to:  
 - Swagger UI: [http://127.0.0.1:8000/docs](http://127.0.0.1:8000/docs)  
 - ReDoc: [http://127.0.0.1:8000/redoc](http://127.0.0.1:8000/redoc)  
  
---  
  
### Example Output  
- \*\*GET `/users/`\*\*: Returns a list of users.  
 ```json  
 {  
 "users": ["Alice", "Bob", "Charlie"]  
 }  
 ```  
  
- \*\*GET `/items/`\*\*: Returns a list of items.  
 ```json  
 {  
 "items": ["Item1", "Item2", "Item3"]  
 }  
 ```  
  
- \*\*POST `/users/`\*\*: Creates a new user.  
 ```json  
 {  
 "message": "User John created successfully!"  
 }  
 ```  
  
- \*\*POST `/items/`\*\*: Creates a new item.  
 ```json  
 {  
 "message": "Item Laptop created successfully!"  
 }  
 ```  
  
---  
  
This structure is scalable and allows you to easily add more routers for different parts of your application.

Here are the \*\*GET\*\* and \*\*POST\*\* URLs you can use to test the above FastAPI application. These URLs correspond to the routes defined in the `users` and `items` routers.  
  
---  
  
### \*\*Base URL\*\*  
Assuming the application is running locally, the base URL is:  
```  
http://127.0.0.1:8000  
```  
  
---  
  
### \*\*Users Router (`/users`)\*\*  
  
#### 1. \*\*GET `/users/`\*\*  
- \*\*Description\*\*: Fetch all users.  
- \*\*URL\*\*:  
 ```  
 http://127.0.0.1:8000/users/  
 ```  
  
#### 2. \*\*GET `/users/{user\_id}`\*\*  
- \*\*Description\*\*: Fetch a specific user by their ID.  
- \*\*Example URL\*\* (replace `{user\_id}` with an actual ID, e.g., `1`):  
 ```  
 http://127.0.0.1:8000/users/1  
 ```  
  
#### 3. \*\*POST `/users/`\*\*  
- \*\*Description\*\*: Create a new user.  
- \*\*URL\*\*:  
 ```  
 http://127.0.0.1:8000/users/  
 ```  
- \*\*Request Body\*\* (JSON):  
 ```json  
 {  
 "name": "John"  
 }  
 ```  
- \*\*Example cURL Command\*\*:  
 ```bash  
 curl -X POST "http://127.0.0.1:8000/users/" -H "Content-Type: application/json" -d '{"name": "John"}'  
 ```  
  
---  
  
### \*\*Items Router (`/items`)\*\*  
  
#### 1. \*\*GET `/items/`\*\*  
- \*\*Description\*\*: Fetch all items.  
- \*\*URL\*\*:  
 ```  
 http://127.0.0.1:8000/items/  
 ```  
  
#### 2. \*\*GET `/items/{item\_id}`\*\*  
- \*\*Description\*\*: Fetch a specific item by its ID.  
- \*\*Example URL\*\* (replace `{item\_id}` with an actual ID, e.g., `1`):  
 ```  
 http://127.0.0.1:8000/items/1  
 ```  
  
#### 3. \*\*POST `/items/`\*\*  
- \*\*Description\*\*: Create a new item.  
- \*\*URL\*\*:  
 ```  
 http://127.0.0.1:8000/items/  
 ```  
- \*\*Request Body\*\* (JSON):  
 ```json  
 {  
 "name": "Laptop"  
 }  
 ```  
- \*\*Example cURL Command\*\*:  
 ```bash  
 curl -X POST "http://127.0.0.1:8000/items/" -H "Content-Type: application/json" -d '{"name": "Laptop"}'  
 ```  
  
---  
  
### \*\*Root Endpoint (`/`)\*\*  
  
#### 1. \*\*GET `/`\*\*  
- \*\*Description\*\*: Fetch the root message.  
- \*\*URL\*\*:  
 ```  
 http://127.0.0.1:8000/  
 ```  
  
---  
  
### Testing with Tools  
  
#### 1. \*\*Using Swagger UI\*\*  
- Open [http://127.0.0.1:8000/docs](http://127.0.0.1:8000/docs) in your browser.  
- You can test all the endpoints directly from the Swagger UI interface.  
  
#### 2. \*\*Using Postman\*\*  
- Import the URLs into Postman.  
- For \*\*POST\*\* requests, set the body to `raw` and use `JSON` format to send the data.  
  
#### 3. \*\*Using cURL\*\*  
- Use the provided cURL commands for testing in the terminal.