

여기서는 다음과 같이 두 개의 템플릿 파일을 만들었다.

- 애플리케이션 홈 페이지용 home.html
- todo 페이지용 todo.html

[그림 4-1]에서 가장 바깥쪽 사각형이 home 템플릿(홈 페이지)이고 안쪽의 작은 사각형이 todo 템플릿(todo 페이지)이다.

템플릿을 만들기 전에 FastAPI 애플리케이션이 Jinja를 사용하도록 설정해보자.

1 todo API 컴포넌트(todo.py)의 POST 라우트를 다음과 같이 수정한다.

```
from fastapi import APIRouter, Path, HTTPException, status, Request, Depends
from fastapi.templating import Jinja2Templates
from model import Todo, TodoItem, TodoItems

todo_router = APIRouter()

todo_list = []

templates = Jinja2Templates(directory="templates/")

@todo_router.post("/todo")
async def add_todo(request: Request, todo: Todo = Depends(Todo.as_form)):
    todo.id = len(todo_list) + 1
    todo_list.append(todo)
    return templates.TemplateResponse("todo.html",
    {
        "request": request,
        "todos": todo_list
    })
```

2 GET 라우트를 다음과 같이 수정한다.

```
@todo_router.get("/todo", response_model=TodoItems)
async def retrieve_todo(request: Request):
    return templates.TemplateResponse("todo.html", {
        "request": request,
        "todos": todo_list
```