

```

    })

    @todo_router.get("/todo/{todo_id}")
    async def get_single_todo(request: Request, todo_id: int = Path(..., title="The ID
    of the todo to retrieve.)):
        for todo in todo_list:
            if todo.id == todo_id:
                return templates.TemplateResponse(
                    "todo.html", {
                        "request": request,
                        "todo": todo
                    })
            raise HTTPException(
                status_code=status.HTTP_404_NOT_FOUND,
                detail="Todo with supplied ID doesn't exist",
            )

```

이 코드는 Jinja가 `template` 폴더를 참조해서 그 안에 있는 특정 템플릿을 사용하도록 지정한다. 템플릿은 `templates.TemplateResponse()` 메서드를 통해 전달된다.

`todo`를 추가하는 `POST` 메서드는 의존성을 사용해서 입력값을 전달한다. 의존성에 관해서는 <CHAPTER 6 데이터베이스 연결>에서 자세히 다룬다.

3 다음 코드에서 강조된 부분을 `model.py`의 `Config` 서브 클래스 앞에 추가한다.

```

from typing import List, Optional
from fastapi import Form

class Todo(BaseModel):
    id: Optional[int]
    item: str

    @classmethod
    def as_form(
        cls,
        item: str = Form(...)
    ):
        return cls(item=item)

```

API 코드를 업데이트해서 Jinja 템플릿을 사용할 수 있게 됐다. 이제 템플릿을 작성해보자.