

Лаб 13 – Автомат тест үүсгэлт (FastAPI Todo API)

Зорилго

Search-based (Pynguin), property-based (Hypothesis), AI/LLM-д суурилсан тест үүсгэлтийн аргуудыг нэг төсөл дээр туршиж харьцуулах.
Coverage хэмжих, илэрсэн асуудлыг тайлагнах, автоматжуулалтын үр ашгийг дүгнэх.

Төслийн сонголт

Сонгосон төсөл: [blog_tutorials/FastAPI_Todo_App/Todo_Part2](#)

Шалтгаан: жижиг CRUD гадаргуу, FastAPI + Pydantic стек, in-memory өгөгдөлтэй тул тестлэх, рефактор хийхэд хурдан.

Онолын товч тайлбар

Search-based testing (Pynguin): Генетик хайлтаар API/функцийн оролт, дуудагдал хослолыг судалж тест үүсгэнэ. Framework-н хамааралгүй цэвэр Python логик дээр хамгийн үр дүнтэй.

Property-based testing (Hypothesis): Олон янзын санамсаргүй оролтыг property-aар тодорхойлж, инвариант/алдааны замыг системтэй илрүүлнэ. Shrinking механизмаар бага жишээгээр алдааг тайлбарлах давуу талтай.

LLM-based testing: Том хэлний загварууд (Copilot/ChatGPT гэх мэт) кодын утгыг тайлбарлан pytest бүтэц, mock/fixtures санал болгодог. Хурдан эхлэл өгөх ч хүний баталгаажуулалт шаардлагатай.

Хийгдсэн ажил

Логикуудыг service.py файлд салгаж, FastAPI-г main.py-д нимгэлж Pynguin-д нийцтэй болгосон.

Pynguin ашиглан service.py дээр smoke тестүүд үүсгэсэн (`tests_pynguin/`).

Hypothesis ашиглан 6 property бичиж CRUD инвариант, алдааны замыг шалгасан (`tests_hypothesis/`).

LLM prompt ашиглан TestClient-тэй API тестүүд бичиж CRUD болон 404 замуудыг хамарсан (`tests_ai/`).

Coverage HTML тайлангуудыг coverage_penguin/html, coverage_hypothesis/html, coverage_ai/html замд хадгалсан.

Лаб тайланг report_Ovog_Ner.docx (өөрийн нэрээр солих) болон README-г монголоор бичсэн.

Тестүүд ба үр дүн

Penguin: service-д ~36% line coverage (smoke).

Hypothesis: service-д ~96% line coverage; KeyError, UUID/ID инвариант, required keys, unique ID.

AI (LLM): main 94%, service 100%, нийт ~96% line coverage; CRUD + 404 замууд TestClient-ээр.

Алдаа, асуудал, шийдэл

Penguin FastAPI/Pydantic дээр pickling алдаа өгсөн → логикийг цэвэр Python service.py болгож салгасан.

In-memory state-ийн нэлэөг багасгахын тулд тест бүрт reset_todos fixture ашигласан.

Windows long-path алдаа гарч болзошгүй → шаардлагатай багцыг салгаж суулгах эсвэл long-path support идэвхжүүлэх.

Ажилгааны хэсгээс

The screenshot shows a Swagger UI interface for a 'Todos' endpoint. The top navigation bar has 'GET /todos/' and 'POST /todos/'. Below the navigation, there are sections for 'Parameters' (No parameters), 'Request body' (required, application/json schema), and 'Example Value'. The example value is a JSON object:

```
{
  "title": "Do home work",
  "description": "Complete all my works of University",
  "completed": false
}
```

Below these sections is a 'Responses' section. Under 'curl', there is a command:

```
curl -X 'POST' \
  'http://127.0.0.1:8000/todos/' \
  -H 'accept: application/json' \
  -H 'Content-type: application/json' \
  -d {
    "title": "Do home work",
    "description": "Complete all my works of University",
    "completed": false
  }
```

Under 'Request URL', the URL is listed as `http://127.0.0.1:8000/todos/`. The 'Server response' section is currently empty.

Code Details

200 Response body

```
{
  "id": "elbce5c5-b330-4a79-af8f-87de37d50736",
  "title": "Do home work",
  "description": "Complete all my works of University",
  "completed": false,
  "created_at": "2025-12-12T22:02:55.807105"
}
```

Download

Response headers

```
content-length: 180
content-type: application/json
date: Fri, 12 Dec 2025 14:02:55 GMT
server: unicorn
```

Responses

| Code | Description | Links |
|------|--|----------|
| 200 | Successful Response | No links |
| | Media type | |
| | application/json | |
| | Controls Accept header. | |
| | Example Value Schema | |
| | { "id": "string", "title": "string", "description": "string", "completed": true, "created_at": "2025-12-12T14:03:11.121Z" } | |
| 422 | Validation Error | No links |
| | Media type | |
| | application/json | |
| | Example Value Schema | |
| | { "detail": [{ "loc": ["id": "string", "title": "string", "description": "string", "completed": true, "created_at": "2025-12-12T14:03:11.121Z"]]] } | |

Name Description

todo_id * required string (path)

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:8000/todos/elbce5c5-b330-4a79-af8f-87de37d50736' \
-H 'accept: application/json'
```

Request URL

http://127.0.0.1:8000/todos/elbce5c5-b330-4a79-af8f-87de37d50736

Server response

| Code | Details |
|------|---|
| 200 | Response body |
| | { "id": "elbce5c5-b330-4a79-af8f-87de37d50736", "title": "Do home work", "description": "Complete all my works of University", "completed": false, "created_at": "2025-12-12T22:02:55.807105" } |
| | Download |
| | Response headers |
| | content-length: 180 content-type: application/json date: Fri, 12 Dec 2025 14:04:00 GMT server: unicorn |
| | Responses |
| Curl | Description |

Дүгнэлт

Property-based (Hypothesis) ба AI тестүүд өндөр coverage, алдааны замуудыг илүү сайн илрүүлсэн.

Search-based (Pynguin) нь framework-салгасан логик дээр л ашигтай; архитектурын салгалт чухал.

Ирээдүйд: core логикийг framework-гүй байлгах, property-based шалгалтыг AI-санаа авч өргөжүүлэх, search-based хэрэгслийг туслах байдлаар ашиглах***