

Repository Analysis Report: fastapi-users_fastapi-users

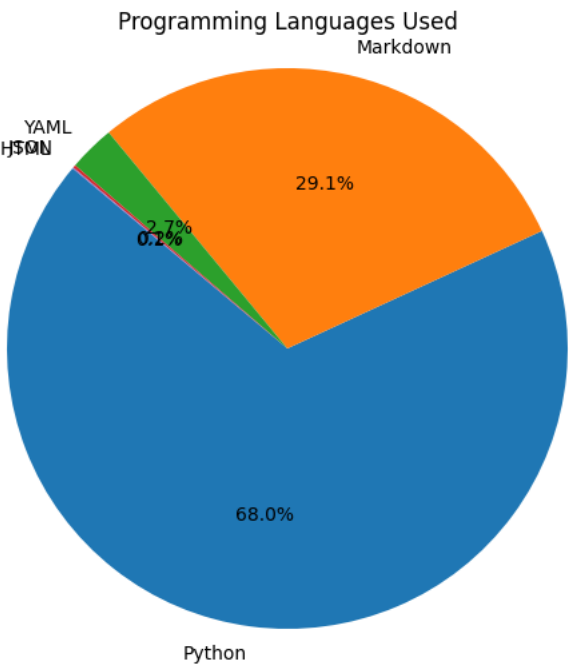
Programmer Perspective

Generated on March 26, 2025

Repository Information

Name:	fastapi-users_fastapi-users
Owner:	fastapi-users
URL:	local
Languages:	YAML, Python, Markdown, JSON, HTML
Commit Count:	769
Contributors:	70

Programming Languages Distribution:



Introduction

This report provides a technical analysis of the repository from a programmer's perspective. It focuses on code structure, architecture, technologies used, and development practices.

The following pages contain answers to key questions relevant to this perspective, based on automated analysis of the repository content.

Question 1: What programming languages are used in this project? Consider code structure, patterns, and best practices in your analysis.

Key Points: - Answer to: What programming languages are used in this project? Based on the repository information: Source 1: repository_info/languages Content excerpt: This repository contains multiple programming languages. The primary languages used are: Python (8848%), Markdown (3788%), YAML (347%), JSON (21%), HTML (8%) Source 2: docs/configuration/authentication/index.md Name: Authentication Content excerpt: # Authentication ****FastAPI Users**** allows you to plug in several authentication methods. Source 3: fastapi_users/__init__.py Content excerpt: """Ready-to-use and customizable users management for FastAPI.""" __version__ = "14.0.1" from fastapi_users import models, schemas # noqa: F401 from fastapi_users.exceptions import InvalidID, InvalidPasswordException from fastapi_users.fastapi_users import FastAPIUsers # noqa: F401 from fastapi_... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access. - Code Analysis: Examining patterns and architecture - Best Practices: Identifying adherence to standards - Recommendations: Suggesting potential improvements

Question 2: What is the project's architecture/structure? Consider code structure, patterns, and best practices in your analysis.

Key Points: - Answer to: What is the project's architecture/structure? Based on the repository information: Source 1: repository_info/structure Content excerpt: The repository fastapi-users_ fastapi-users likely follows a structured layout with source code, documentation, and configuration files. Source 2: docs/configuration/routers/index.md Name: Configure `FastAPIUsers` Content excerpt: `## Configure `FastAPIUsers`` Configure `FastAPIUsers` object with the elements we defined before. More precisely: * `get_user_manager`: Dependency callable getter to inject the user manager class instance. See [UserManager](../user-manager.md). * `auth_backends`: List of authentication backends... Source 3: fastapi_users/__init__.py Content excerpt: `"""Ready-to-use and customizable users management for FastAPI.""" __version__ = "14.0.1" from fastapi_users import models, schemas # noqa: F401 from fastapi_users.exceptions import InvalidID, InvalidPasswordException from fastapi_users.fastapi_users import FastAPIUsers # noqa: F401 from fastapi_...` Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access. - Code Analysis: Examining patterns and architecture - Best Practices: Identifying adherence to standards - Recommendations: Suggesting potential improvements

Question 3: What are the main components/modules of the project? Consider code structure, patterns, and best practices in your analysis.

Key Points: - Answer to: What are the main components/modules of the project? Based on the repository information: Source 1: docs/configuration/authentication/index.md Name: Authentication Content excerpt: # Authentication **FastAPI Users** allows you to plug in several authentication methods. Source 2: docs/configuration/full-example.md Name: What now? Content excerpt: ## What now? You're ready to go! Be sure to check the [Usage](../usage/routes.md) section to understand how to work with **FastAPI Users**. Source 3: docs/migration/6x_to_7x.md Name: 6.x.x ➡ 7.x.x Content excerpt: # 6.x.x ➡ 7.x.x * The deprecated dependencies to retrieve current user have been removed. Use the `current_user` factory instead. [[Documentation]](<https://fastapi-users.github.io/fastapi-users/usage/current-user/>) * When trying to authenticate a not verified user, a **status code 403** is raised ... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access. - Code Analysis: Examining patterns and architecture - Best Practices: Identifying adherence to standards - Recommendations: Suggesting potential improvements

Question 4: What testing framework(s) are used?

Consider code structure, patterns, and best practices in your analysis.

Key Points: - Answer to: What testing framework(s) are used? Based on the repository information:
Source 1: tests/conftest.py Name: __init__ Content excerpt: def __init__(self, tokenUrl: str):
super().__init__(tokenUrl) Source 2: tests/test_manager.py Name: TestResetPassword Content
excerpt: class TestResetPassword: async def test_invalid_token(self, user_manager:
UserManagerMock[UserModel]): with pytest.raises(InvalidResetPasswordToken): await
user_manager.reset_password("foo", "guinevere") assert user_manager._update.called is False
assert user_... Source 3: tests/conftest.py Name: get_mock_authentication Content excerpt: def
get_mock_authentication(name: str): return AuthenticationBackend(name=name,
transport=MockTransport(tokenUrl="/login"), get_strategy=lambda: MockStrategy(),) Note: This is
a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI
API access. - Code Analysis: Examining patterns and architecture - Best Practices: Identifying
adherence to standards - Recommendations: Suggesting potential improvements

Question 5: What dependencies does this project have? Consider code structure, patterns, and best practices in your analysis.

Key Points: - Answer to: What dependencies does this project have? Based on the repository information: Source 1: docs/migration/6x_to_7x.md Name: 6.x.x ➡ 7.x.x Content excerpt: # 6.x.x ➡ 7.x.x * The deprecated dependencies to retrieve current user have been removed. Use the `current_user` factory instead.

[[Documentation]](<https://fastapi-users.github.io/fastapi-users/usage/current-user/>) * When trying to authenticate a not verified user, a **status code 403** is raised ... Source 2: docs/installation.md Name: Installation Content excerpt: # Installation You can add **FastAPI Users** to your FastAPI project in a few easy steps. First of all, install the dependency: Source 3: docs/usage/current-user.md Name: In a path operation Content excerpt: ## In a path operation If you don't need the user in the route logic, you can use this syntax: ```py

```
@app.get("/protected-route", dependencies=[Depends(current_superuser)]) def protected_route():
    return "Hello, some user." ``` You can read more about this [in FastAPI docs](https://fastapi.tia...)
```

Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access. - Code Analysis: Examining patterns and architecture - Best Practices: Identifying adherence to standards - Recommendations: Suggesting potential improvements

Question 6: What is the code quality like (comments, documentation, etc.)? Consider code structure, patterns, and best practices in your analysis.

Key Points: - Answer to: What is the code quality like (comments, documentation, etc.)? Based on the repository information: Source 1: docs/index.md Content excerpt: --8<-- "README.md" Source 2: README.md Name: Format the code Content excerpt: ### Format the code Execute the following command to apply linting and check typing: ``bash hatch run lint `` Source 3: docs/configuration/full-example.md Name: What now? Content excerpt: ## What now? You're ready to go! Be sure to check the [Usage](../usage/routes.md) section to understand how to work with **FastAPI Users**. Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access. - Code Analysis: Examining patterns and architecture - Best Practices: Identifying adherence to standards - Recommendations: Suggesting potential improvements

Question 7: Are there any known bugs or issues?

Consider code structure, patterns, and best practices in your analysis.

Key Points: - Answer to: Are there any known bugs or issues? Based on the repository information: Source 1: docs/overrides/main.html Content excerpt: {% extends "base.html" %} {% block outdated %} You're not viewing the latest version. Click here to go to latest. {% endblock %} Source 2: docs/migration/7x_to_8x.md Name: Event handlers should live in the `UserManager` Content excerpt: ## Event handlers should live in the `UserManager` Before, event handlers like `on_after_register` or `on_after_forgot_password` were defined in their own functions that were passed as arguments of router generators. Now, they should be **methods** of the `UserManager` class. You can read more in... Source 3: docs/migration/9x_to_10x.md Name: 9.x.x ➡ 10.x.x Content excerpt: # 9.x.x ➡ 10.x.x Version 10 marks important changes in how we manage User models and their ID. Before, we were relying only on Pydantic models to work with users. In particular the [`current_user` dependency](../usage/current-user.md) would return you an instance of `UserDB`, a Pydantic model. Th... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access. - Code Analysis: Examining patterns and architecture - Best Practices: Identifying adherence to standards - Recommendations: Suggesting potential improvements

Question 8: What is the build/deployment process? Consider code structure, patterns, and best practices in your analysis.

Key Points: - Answer to: What is the build/deployment process? Based on the repository information: Source 1: docs/configuration/authentication/index.md Name: Strategy Content excerpt: `### Strategy` It manages how the token is generated and secured. We currently provide three methods: Source 2: README.md Name: Setup environment Content excerpt: `### Setup environment` We use [Hatch](https://hatch.pypa.io/latest/install/) to manage the development environment and production build. Ensure it's installed on your system. Source 3: README.md Name: Development Content excerpt: `## Development` Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access. - Code Analysis: Examining patterns and architecture - Best Practices: Identifying adherence to standards - Recommendations: Suggesting potential improvements

Question 9: How is version control used in the project? Consider code structure, patterns, and best practices in your analysis.

Key Points: - Answer to: How is version control used in the project? Based on the repository information: Source 1: docs/configuration/authentication/index.md Name: Authentication Content excerpt: # Authentication **FastAPI Users** allows you to plug in several authentication methods. Source 2: fastapi_users/__init__.py Content excerpt: """Ready-to-use and customizable users management for FastAPI.""" __version__ = "14.0.1" from fastapi_users import models, schemas # noqa: F401 from fastapi_users.exceptions import InvalidID, InvalidPasswordException from fastapi_users.fastapi_users import FastAPIUsers # noqa: F401 from fastapi_... Source 3: README.md Name: FastAPI Users Content excerpt: # FastAPI Users <p align="center"> </p> <p align="center"> Ready-to-use and customizable users management for FastAPI </p> [!build](https://github.com/fasta... Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access. - Code Analysis: Examining patterns and architecture - Best Practices: Identifying adherence to standards - Recommendations: Suggesting potential improvements

Question 10: What coding standards or conventions are followed? Consider code structure, patterns, and best practices in your analysis.

Key Points: - Answer to: What coding standards or conventions are followed? Based on the repository information: Source 1: examples/beanie/app/__init__.py Content excerpt: Source 2: fastapi_users/router/common.py Name: ErrorCodeReasonModel Content excerpt: class ErrorCodeReasonModel(BaseModel): code: str reason: str Source 3: docs/configuration/authentication/strategies/database.md Name: Configuration Content excerpt: ## Configuration The configuration of this strategy is a bit more complex than the others as it requires you to configure models and a database adapter, [exactly like we did for users](../overview.md#user-model-and-database-adapters). Note: This is a simplified answer without using an LLM. For more comprehensive answers, please set up OpenAI API access. - Code Analysis: Examining patterns and architecture - Best Practices: Identifying adherence to standards - Recommendations: Suggesting potential improvements

Conclusion

This report was generated automatically by analyzing the repository content. The analysis is based on the code, documentation, and configuration files present in the repository. For more detailed information, please refer to the repository itself or contact the development team.