

Trilha 06

*Documentação Automatizada e API
Docs*

Instruções para a melhor prática de Estudo

1. **Leia atentamente todo o conteúdo:** Antes de iniciar qualquer atividade, faça uma leitura detalhada do material fornecido na trilha, compreendendo os conceitos e os exemplos apresentados.
2. **Não se limite ao material da trilha:** Utilize o material da trilha como base, mas busque outros materiais de apoio, como livros, artigos acadêmicos, vídeos, e blogs especializados. Isso enriquecerá o entendimento sobre o tema.
3. **Explore a literatura:** Consulte livros e publicações reconhecidas na área, buscando expandir seu conhecimento além do que foi apresentado. A literatura acadêmica oferece uma base sólida para a compreensão de temas complexos.
4. **Realize todas as atividades propostas:** Conclua cada uma das atividades práticas e teóricas, garantindo que você esteja aplicando o conhecimento adquirido de maneira ativa.
5. **Evite o uso de Inteligência Artificial para resolução de atividades:** Utilize suas próprias habilidades e conhecimentos para resolver os exercícios. O aprendizado vem do esforço e da prática.
6. **Participe de debates:** Discuta os conteúdos estudados com professores, colegas e profissionais da área. O debate enriquece o entendimento e permite a troca de diferentes pontos de vista.
7. **Pratique regularmente:** Não deixe as atividades para a última hora. Pratique diariamente e revise o conteúdo com frequência para consolidar o aprendizado.
8. **Peça feedback:** Solicite o retorno dos professores sobre suas atividades e participe de discussões sobre os erros e acertos, utilizando o feedback para aprimorar suas habilidades.

Essas instruções são fundamentais para garantir um aprendizado profundo e eficaz ao longo das trilhas.

Documentação Automatizada e API Docs

A documentação automatizada é uma prática que simplifica a criação e manutenção de documentação técnica, gerando conteúdo diretamente a partir do código-fonte. Essa abordagem é particularmente útil para a documentação de APIs, onde as definições e comportamentos dos endpoints podem ser gerados e atualizados automaticamente. O MKDocs pode ser integrado com ferramentas que suportam a geração automática de documentação, fornecendo uma solução eficiente e de alta qualidade.

Introdução à Documentação Automatizada de Código

A documentação automatizada envolve a geração de documentos a partir de comentários no código ou de ferramentas que interpretam o código-fonte. Isso evita a necessidade de escrever toda a documentação manualmente, reduzindo o risco de discrepâncias entre o código e a documentação.

Vantagens da Documentação Automatizada:

- **Redução de Erros:** A documentação gerada automaticamente reflete com precisão o código atual, evitando desatualizações.
- **Economia de Tempo:** Desenvolvedores gastam menos tempo escrevendo e mantendo documentação manual.
- **Manutenção Simplificada:** A documentação é atualizada junto com o código, garantindo consistência.

Ferramentas Comuns para Documentação Automatizada:

- **Sphinx:** Uma ferramenta de documentação para projetos Python, capaz de gerar documentação automaticamente a partir de docstrings.
- **Javadoc:** Gera documentação para projetos Java, baseado nos comentários em docstrings.
- **Doxygen:** Usado para gerar documentação de projetos C++, C, Java, entre outros.
- **Swagger/OpenAPI:** Muito utilizado para documentar APIs RESTful, gerando interfaces interativas a partir da definição da API.

Integração do MKDocs com Ferramentas de Geração de Documentação Automática

O MKDocs pode ser integrado a várias ferramentas de geração automática de documentação para simplificar a criação de documentação técnica, especialmente de APIs.

Exemplo: Integração com o Swagger/OpenAPI

O **Swagger** é uma ferramenta amplamente usada para gerar documentação de APIs a partir da especificação OpenAPI. O Swagger não só documenta as APIs, mas também gera uma interface interativa onde os usuários podem testar os endpoints diretamente do navegador.

Passo a Passo para Integração com MKDocs:

1. **Instale o Swagger-UI como uma Dependência:** O Swagger-UI pode ser utilizado para criar uma interface de documentação interativa dentro do MKDocs. Para integrá-lo:
 - Baixe e adicione os arquivos do Swagger-UI ao diretório de documentação do MKDocs.

1. **Criação da Definição da API com OpenAPI:** Crie um arquivo YAML ou JSON que descreva sua API conforme a especificação OpenAPI. Exemplo de definição básica em YAML:

```
openapi: 3.0.0
info:
  title: API de Exemplo
  description: Exemplo de API para documentação automatizada
  version: 1.0.0
paths:
  /produtos:
    get:
      summary: Lista todos os produtos
      responses:
        '200':
          description: Sucesso
```

2. **Configuração no `mkdocs.yml`:** Adicione o Swagger-UI ao projeto e inclua a especificação OpenAPI.

```
site_name: 'Documentação da API'
theme:
  name: 'material'
plugins:
  - search
nav:
  - Home: index.md
  - API: api.md
```

3. **Página da API:** No arquivo `api.md`, insira um `iframe` para carregar a interface Swagger.

```
# API Documentação

<iframe src="path/to/swagger-ui.html" width="100%"
height="800"></iframe>
```

4. **Build e Deploy:** Após configurar o Swagger, rode o comando para gerar e publicar a documentação:

```
mkdocs build

mkdocs serve
```

5. Com essas etapas, você pode integrar a documentação automatizada da API dentro do MKDocs, proporcionando uma visualização interativa e precisa.

Documentação de APIs com Exemplos Práticos

Documentar APIs corretamente é fundamental para garantir que desenvolvedores externos e internos consigam interagir com o sistema de forma eficiente. Exemplos práticos ajudam os usuários a entenderem como consumir a API.

Exemplo de Documentação de API:

```
openapi: 3.0.0
info:
  title: API de Produto
  description: API para gerenciar produtos em um sistema de
e-commerce
  version: 1.0.0
paths:
  /produtos:
    get:
      summary: Lista todos os produtos
      responses:
        '200':
          description: Sucesso
          content:
            application/json:
              schema:
                type: array
                items:
                  type: object
                  properties:
                    id:
                      type: integer
                      example: 1
                    nome:
                      type: string
                      example: "Cadeira de Escritório"
                    preco:
                      type: number
                      example: 499.99
    post:
      summary: Cria um novo produto
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                nome:
                  type: string
                  example: "Mesa de Jantar"
```

```
    preco:
      type: number
      example: 699.99
  responses:
    '201':
      description: Produto criado com sucesso
```

Nesse exemplo, a documentação especifica dois endpoints: um para listar produtos e outro para criar novos produtos. Além disso, exemplos práticos de requisição e resposta são fornecidos, o que facilita o entendimento do funcionamento da API.

Lista de Exercícios de Fixação

1. **Exercício 1:** Explique as vantagens da documentação automatizada de código e como ela pode beneficiar uma equipe de desenvolvimento.
2. **Exercício 2:** Crie um pequeno projeto que utilize o MKDocs integrado ao Swagger para documentar uma API de um sistema fictício de gerenciamento de usuários.
3. **Exercício 3:** Usando a especificação OpenAPI, documente uma API com pelo menos três endpoints: um para listar, um para criar e um para atualizar recursos. Gere e visualize a documentação com o MKDocs.
4. **Exercício 4:** Pesquise sobre outra ferramenta de documentação automatizada (como Doxygen ou Sphinx) e explique como ela pode ser integrada ao MKDocs.
5. **Exercício 5:** Modifique o exemplo fornecido da API de produtos, adicionando um endpoint para excluir um produto, e atualize a documentação automaticamente com o Swagger.