

YAML e Arquivos de Logs em APIs de Logging

QXD0099 - Desenvolvimento de Software para Persistência

Universidade Federal do Ceará - *Campus* Quixadá

Prof. Francisco Victor da Silva Pinheiro
victorpinheiro@ufc.br



Agenda

- YAML (YAML Ain't Markup Language)
- Características
- Exemplo de Configuração YAML
- Arquivos de Logs
- Formato Comum de Logs
- Principais Elementos de um Log
- APIs de Logging
- Plataformas de Logging Centralizado
- YAML + Logging

YAML (YAML Ain't Markup Language)

- YAML é uma linguagem de serialização de dados legível para humanos, usada para configuração de aplicativos, especialmente em ambientes como DevOps, Kubernetes e APIs.
- O YAML é usado em vários projetos, como:
 - Ansible, para criar processos de automação
 - Kubernetes, para implantações e recursos
 - DevOps, para gerir contentores no pipeline
 - Infraestrutura como código (IaC), para definir e gerenciar a infraestrutura de TI



Características

- Estrutura hierárquica:
 - Usa indentação para organizar dados (não usa {}, [] como JSON).
- Suporta múltiplos tipos de dados:
 - Strings, números, booleanos, listas e objetos.
- Legibilidade:
 - Fácil de ler e escrever para humanos.



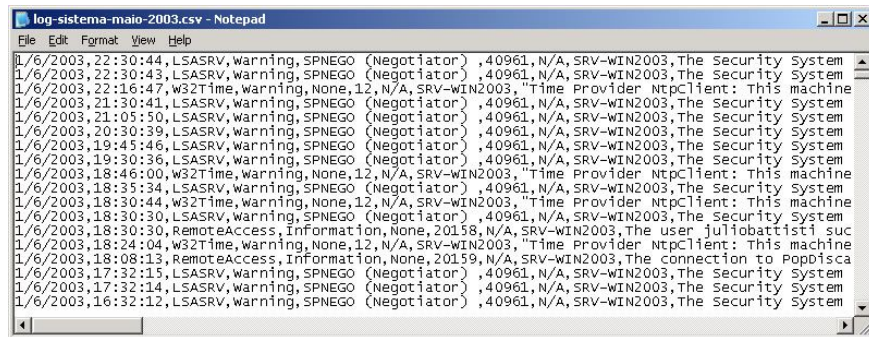
Exemplo de Configuração YAML

```
logging:
  level: INFO
  file: /var/log/app.log
  format: "%(asctime)s - %(name)s - %(levelname)s - %(message)s"
```

- logging.level: Define o nível de log (INFO, DEBUG, ERROR).
- logging.file: Especifica o arquivo onde os logs serão armazenados.
- logging.format: Configura o formato da mensagem do log.

Arquivos de Logs

- Logs são registros que capturam eventos e mensagens gerados por um sistema ou aplicação. Eles são usados para:
 - Monitoramento: Acompanhar o estado do sistema.
 - Depuração: Identificar e corrigir erros.
 - Auditoria: Registrar eventos importantes, como autenticações e alterações.



```

log-sistema-maio-2003.csv - Notepad
File Edit Format View Help
1/6/2003,22:30:44,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The Security System
1/6/2003,22:30:43,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The Security System
1/6/2003,22:16:47,W32Time,warning,None,12,N/A,SRV-WIN2003,"Time Provider NtpClient: This machine
1/6/2003,21:30:41,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The Security System
1/6/2003,21:05:50,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The Security System
1/6/2003,20:30:39,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The Security System
1/6/2003,19:45:46,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The security system
1/6/2003,19:30:36,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The Security System
1/6/2003,18:46:00,W32Time,warning,None,12,N/A,SRV-WIN2003,"Time Provider NtpClient: This machine
1/6/2003,18:35:34,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The security system
1/6/2003,18:30:44,W32Time,warning,None,12,N/A,SRV-WIN2003,"Time Provider NtpClient: This machine
1/6/2003,18:30:30,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The security system
1/6/2003,18:30:30,RemoteAccess,Information,None,20158,N/A,SRV-WIN2003,The user juliobattisti suc
1/6/2003,18:24:04,W32Time,warning,None,12,N/A,SRV-WIN2003,"Time Provider NtpClient: This machine
1/6/2003,18:08:13,RemoteAccess,Information,None,20159,N/A,SRV-WIN2003,The connection to Popd1sca
1/6/2003,17:32:15,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The Security System
1/6/2003,17:32:14,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The Security System
1/6/2003,16:32:12,LSASRV,warning,SPNEGO (Negotiator),40961,N/A,SRV-WIN2003,The Security System
    
```

Formato Comum de Logs

- Os logs geralmente seguem um formato padrão, como:
 - *[2024-11-20 12:00:00] INFO - Application started*
 - *[2024-11-20 12:01:00] ERROR - Database connection failed*

- **Principais Elementos de um Log**
 - Timestamp: Data e hora do evento.
 - Nível de log: Severidade do evento (DEBUG, INFO, WARNING, ERROR, CRITICAL).
 - Mensagem: Texto descritivo do evento.

APIs de Logging

- APIs de Logging são interfaces que permitem coletar, armazenar e analisar logs de forma programática.
- Bibliotecas Comuns para Logging
- Python - logging:
 - Biblioteca padrão para gerenciar logs.
 - Configuração básica

```
import logging

logging.basicConfig(level=logging.INFO, filename='app.log',
                    format='%(asctime)s - %(levelname)s - %(message)s')

logging.info("App iniciado")
logging.error("Erro no banco de dados")
```


APIs de Logging

- Java - SLF4J com Logback:
 - Configuração via XML ou YAML

```
<configuration>
  <appender name="FILE" class="ch.qos.logback.core.FileAppender">
    <file>logs/app.log</file>
    <encoder>
      <pattern>%d{yyyy-MM-dd HH:mm:ss} %-5level %logger{36} - %msg%n</pattern>
    </encoder>
  </appender>
  <root level="info">
    <appender-ref ref="FILE" />
  </root>
</configuration>
```

APIs de Logging

- Node.js - Winston:
 - Um popular gerenciador de logs

```
const winston = require('winston');

const logger = winston.createLogger({
  level: 'info',
  format: winston.format.json(),
  transports: [
    new winston.transports.File({ filename: 'app.log' })
  ]
});

logger.info('App iniciado');
logger.error('Erro no banco de dados');
```

Plataformas de Logging Centralizado

- APIs modernas frequentemente usam plataformas como:
 - ELK Stack (Elasticsearch, Logstash, Kibana): Para centralizar e analisar logs.
 - Graylog: Solução de logging centralizado.
 - Cloud Logging (Google), AWS CloudWatch Logs, Azure Monitor: Ferramentas nativas para nuvem.

YAML + Logging

- YAML é frequentemente usado para configurar sistemas de logging. Exemplo:

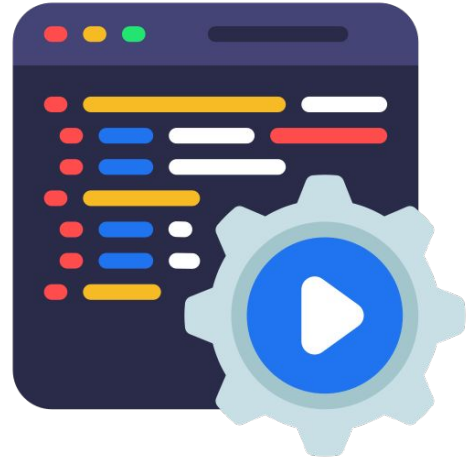
```
handlers:
  file_handler:
    class: logging.FileHandler
    level: DEBUG
    formatter: detailed
    filename: app.log

formatters:
  detailed:
    format: "%(asctime)s - %(name)s - %(levelname)s - %(message)s"

loggers:
  my_logger:
    level: DEBUG
    handlers: [file_handler]
```

Referências

- SMITH, Ben. JSON básico: conheça o formato de dados preferido da web. São Paulo: Novatec, 2015.
- <https://en.wikipedia.org/wiki/JSON>
- https://www.w3schools.com/js/js_json_intro.asp
- <https://spacetelescope.github.io/understanding-json-schema/>
- <http://json-schema.org/>
- <https://github.com/FasterXML/jackson-databind>
- <http://www.mkyong.com/java/jackson-2-convert-java-object-to-from-json/>
- Jackson vs Gson (com exemplos)
 - <http://www.baeldung.com/jackson-vs-gson>
 -



Obrigado!

Dúvidas?



Universidade Federal do Ceará - *Campus* Quixadá

Prof. Francisco Victor da Silva Pinheiro
victorpinheiro@ufc.br

