

File: .pdfignore.yml

```
directories:
  - .git
  - log
  - public/system
  - tmp
  - vendor
files:
  - .DS_Store
  - database.yml
  - favicon.ico
patterns:
  - '*.pdf'
  - '*.tmp'
# - '*.py'
```

File: code-to-pdf.py

```
import os
import yaml
import markdown
import fnmatch
from reportlab.lib.pagesizes import letter
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
from reportlab.pdfbase import pdfmetrics
from reportlab.pdfbase.cidfonts import UnicodeCIDFont

def load_exclusions(config_path):
    with open(config_path, 'r', encoding='utf-8') as file:
        config = yaml.safe_load(file)
        excluded_dirs = set(config.get('directories', []))
        excluded_files = set(config.get('files', []))
        excluded_patterns = config.get('patterns', [])
        return excluded_dirs, excluded_files, excluded_patterns

def generate_pdf(source_folder, output_pdf, config_path):
    excluded_dirs, excluded_files, excluded_patterns = load_exclusions(config_path)
    doc = SimpleDocTemplate(output_pdf, pagesize=letter)
    story = []
    styles = getSampleStyleSheet()
    pdfmetrics.registerFont(UnicodeCIDFont('STSong-Light'))
    code_style = ParagraphStyle('Code', parent=styles['Normal'], fontName='STSong-Light', fontSi
ze=10, leading=12)
    for root, dirs, files in os.walk(source_folder, topdown=True):
        dirs[:] = [d for d in dirs if d not in excluded_dirs]
        for filename in files:
            filepath = os.path.join(root, filename)
            if filename not in excluded_files and not any(fnmatch.fnmatch(filepath,
os.path.join(source_folder, pattern)) for pattern in excluded_patterns):
                with open(filepath, 'r', encoding='utf-8') as file:
                    content = file.read()
                    if filename.endswith('.md'):
```

```

        html = markdown.markdown(content)
        content = html

        story.append(Paragraph(f"File: {filename}", styles['Title']))
        for line in content.splitlines():
            story.append(Paragraph(line.replace(' ', ' '), co
de_style))

        story.append(Spacer(1, 12))

    doc.build(story)
# 使用该函数
generate_pdf('./', 'output_code.pdf', '.pdfignore.yml')

```

File: test.py

```

import os
from reportlab.lib.pagesizes import letter
from reportlab.platypus import SimpleDocTemplate, Paragraph, Spacer
from reportlab.lib.styles import getSampleStyleSheet, ParagraphStyle
def generate_pdf(source_folder, output_pdf):
    doc = SimpleDocTemplate(output_pdf, pagesize=letter)
    story = []
    styles = getSampleStyleSheet()
    code_style = ParagraphStyle('Code', parent=styles['Normal'], fontName='Courier', fontSize=10
, leading=12)
    for filename in sorted(os.listdir(source_folder)):
        if filename.endswith(".py"): # 根据需要调整文件类型
            filepath = os.path.join(source_folder, filename)
            with open(filepath, 'r', encoding='utf-8') as file:
                code = file.read()
                story.append(Paragraph(f"File: {filename}", styles['Title']))
                for line in code.splitlines():
                    story.append(Paragraph(line.replace(' ', ' '), code_style))
                story.append(Spacer(1, 12))

    doc.build(story)
# 使用该函数
generate_pdf('./', 'output_code2.pdf')

```

File: README.md

Code-to-one-PDF