






Java From a custom data provider [HOME](#)  [GUIDES](#)  [JAVA](#)  [OPEN A DOCUMENT](#)  [FROM CUSTOM DATA PROVIDER](#)

Open PDFs from custom data providers in Java



COPY PAGE



This guide explains how to open a PDF document from a custom data provider.

`PdfDocument` expects a `DataProvider` to read from. You can implement an `InputStreamDataProvider` (which is a `DataProvider`) to provide data from any custom source.

It's also useful to extend a `WritableDataProvider` in the same class to support writing back to the custom data source.

For example, here's a `CustomProvider` that takes a `byte` array as the input data on construction:

```
// A custom data provider that extends `InputStreamDataProvider` and implements `WritableDataProvider`  
// so you can read and write to a data block, `pdfData`.  
class CustomProvider extends InputStreamDataProvider implements WritableDataProvider {  
    private List<Byte> pdfData = new ArrayList<>();  
  
    CustomProvider(@NotNull final byte[] data) {  
        for (byte value : data) {  
            pdfData.add(value);  
        }  
    }  
}
```



ASK AI

```

    }
}

@NotNull
@Override
protected InputStream openInputStream() {
    byte[] data = new byte[pdfData.size()];
    for (int i = 0; i < pdfData.size(); i++) {
        data[i] = pdfData.get(i);
    }
    return new ByteArrayInputStream(data);
}

@Override
public long getSize() {
    return pdfData.size();
}

@Nullable
@Override
public String getTitle() {
    return "Fake-Title";
}

@Override
public boolean canWrite() {
    // You're allowing writing.
    return true;
}

@Override
public boolean startWrite(WriteMode writeMode) {
    // Remove the current data if you're rewriting from scratch.
    if (writeMode == WriteMode.REWRITE) {
        pdfData.clear();
    }

    return true;
}

@Override
public boolean write(byte[] bytes) {
    for (byte value : bytes) {
        pdfData.add(value);
    }
    return true;
}

```

```

    }

    @Override
    public boolean finishWrite() {
        // Nothing to do.
        return true;
    }

    @Override
    public boolean supportsAppending() {
        // You can append onto the string if you want to.
        return true;
    }
}

```

Here's an example of loading a `PdfDocument` using the `CustomProvider` provider above by reading the bytes from a PDF file:

```

{
    FileInputStream fileInputStream = new FileInputStream(new File("document.pdf"));
    int numberOfBytesToRead = fileInputStream.available();
    byte[] buffer = new byte[numberOfBytesToRead];
    fileInputStream.read(buffer, 0, numberOfBytesToRead);
    PdfDocument document = PdfDocument.open(new CustomProvider(buffer));
}

```

Another example of this can be found in the `ReadWriteCustomProviderExample` Catalog example.

Was this helpful?

✓ YES

✗ NO



Copyright 2025 Nutrient. All rights reserved.