

A dark, atmospheric photograph of a mining operation. In the background, a large, steep, rocky cliff face is visible. In the foreground and middle ground, several yellow and orange mining vehicles are active. A large yellow excavator with the number '107' is prominent in the upper center. Below it, another excavator with the number '105' is visible. In the lower left, a large yellow haul truck is partially shown. In the lower right, another haul truck with the number '261' is visible. The overall scene is dimly lit, with the primary light source being the vehicles' headlights and the ambient light from the sky, creating a sense of scale and industrial activity.

MINING ELECTRONIC DOCUMENTS

FOR FUN AND PROFIT

(AND OTHER BUSINESS CRITICAL NEEDS)

README.TXT

- Raymond Camden
- Senior Developer Evangelist for Adobe
- raymondcamden.com
- @raymondcamden (DMs are open!)
- @raymondcamden@mastodon.social





GOOD NEWS

- PDFs are easy to store
- PDFs are easy to share
- PDFs are awesome (no, really!)

BAD NEWS

- Estimated 2.5 trillion PDFs
- According to highly paid consultants, 2.5T is a lot
- Data is safe, but "hidden" in documents

OUR GOAL



Yes, this is a completely gratuitous slide.

HOW?

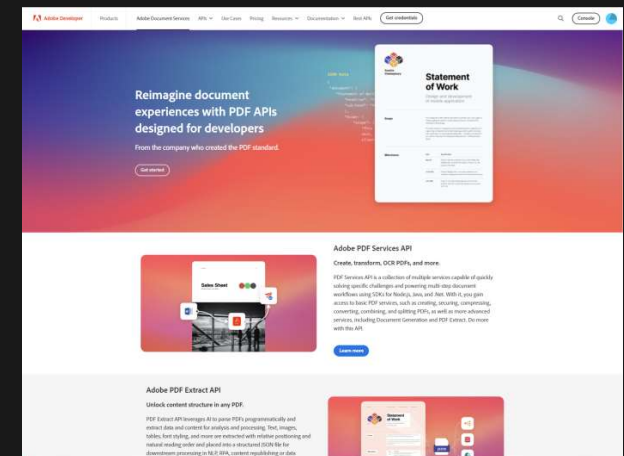
- Get "information" from the PDFs
- Analyze that "information"

GETTING STUFF FROM PDFS

- Text
- Styling Information
- Tables
- Images

SOLUTION

- Adobe Document Services
- APIs related to Documents (duh)
- Created by PDF Wizards



ADOBE DOCUMENT SERVICES

- PDF Services
- Document Generation
- PDF Embed
- Acrobat Sign
- PDF Extract




ADOBE PDF EXTRACT

- Uses Adobe Sensei (ML, AI, Skynet, etc)
- Extracts text, tables, images, styling information
- Tables can be CSV, XLSX, or images
- Extracts document structure
- Auto OCRs when necessary

DETAILS

- SDKs for Node, Java, .NET, Python^{*}
- REST API ✨
- Free trial (1K calls over 6 months)

CODE PROCESS

1. Get credentials (one time)
2. Get the SDK (or use REST)
3. Write code to extract crap from PDF x
4. Automate the previous step
5. Profit   

There may be a few steps missing between 4 and 5.

FOR TODAY

- Node.js ("Node.js Is Bad Ass Rock Star Tech")
- Consider REST

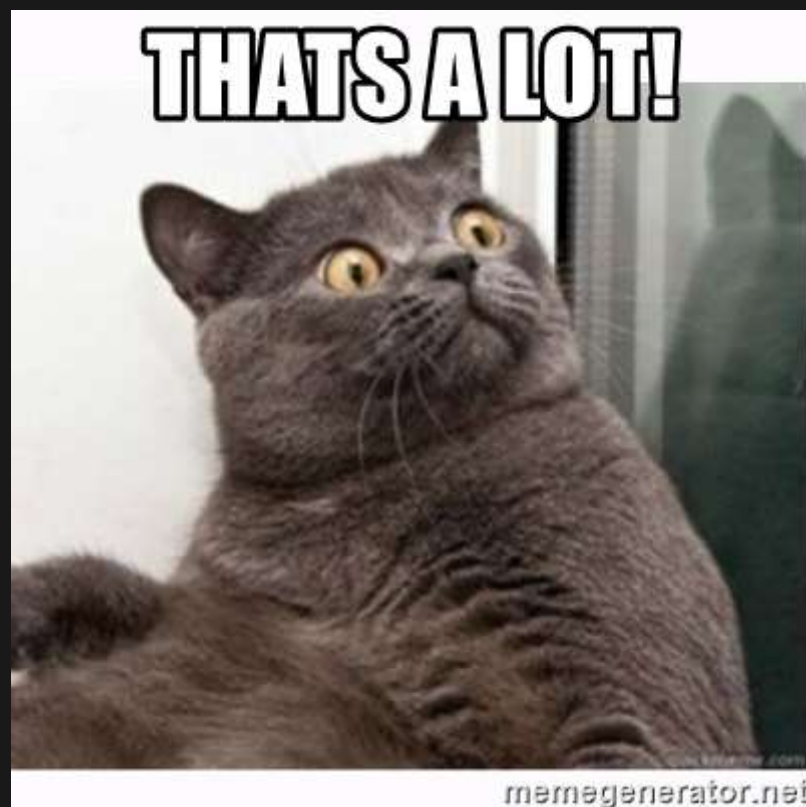
CODE TIME!



GENERAL PSEUDO-CODE FLOW

```
make a credentials object  
create an execution context specific to your operation  
    (extract pdf, ocr pdf, etc)  
set your input and options  
execute  
save result
```

Ray, show /demos/extract/extract.js



WHAT IT MEANS

- Docs
- JSONSchema (ray, if you didn't click it in the previous link, do so)
- Visualizer

NOW WHAT?

I don't know - thanks for showing up - any questions?

SCENARIO - GET TEXT

- Use for search engine
- Use for fragment

Ray, show /demos/extract/text.js

SCENARIO - GET HEADERS

- "Auto" summary
- Search (again)

Ray, show /demos/extract/headers.js

SCENARIO - STYLE COMPLIANCE

- Look for fonts
- Look for text size issues

Ray, show /demos/check_fonts.js

SCENARIO - TEXT COMPLIANCE

- Look for words we don't want
- Look for words that must include others

Ray, show /demos/check_text.js

SCENARIO - PROCESS TABULAR DATA

- Analyze data as data
- Analytics, averages, etc etc

Ray, show /demos/extract/get_data_tables.js and
process_data_tables.js

The image features two vintage-style volume knobs, likely from a car stereo or vintage radio, set against a dark, textured background. The knobs are metallic with a brushed finish and have a circular design with a central screw. Above each knob, the word "VOLUME" is faintly visible. The knobs are positioned on either side of the central text. The text "LET'S TURN IT UP" is written in a bold, white, sans-serif font, centered horizontally and vertically. The overall mood is nostalgic and suggests a theme of increasing volume or intensity.

LET'S TURN IT UP

ML/AI/ETC

- What is the text discussing?
- Who is the text discussing?
- Legal clauses
- Problematic language

SCENARIO - SENTIMENT OF A DOCUMENT

- Incoming docs from customer service reports
- Review information on products
- Diffbot

Ray, show /demos/extract/sentiment.js

SCENARIO - FACTS

- What statements are made in the document?
- Fact !== Truth

Ray, show /demos/extract/facts

SCENARIO - ENTITIES

- What's "things" are discussed?
- People, places, organizations
- "Adobe" != Mud
- If document discusses X, flag it, move it, etc

Ray, show /demos/extract/entities.js

SCENARIO - SUMMARIZE

- What's the gist of a document?
- Let people quickly browse PDFs
- MeaningCloud

Ray, show /demos/extract/summarize.js

SCENARIO - IMAGE ANALYSIS

- Associate picture data with PDF
- Flag problematic images
- Microsoft Computer Vision

Ray, show /computer_vision, analyze then gather

Blog post

RESOURCES

- Docs
- Support Forum
- StackOverflow tags: [adobe-documentgeneration](#), [adobe-embed-api](#), [adobe-pdfservices](#)
- Blog

QUESTIONS?

