Introduction into APIs

Web-scraping and APIs help us to collect info

Web-scraping: automated collection of information directly from web pages, that may or may not be intended for that use.

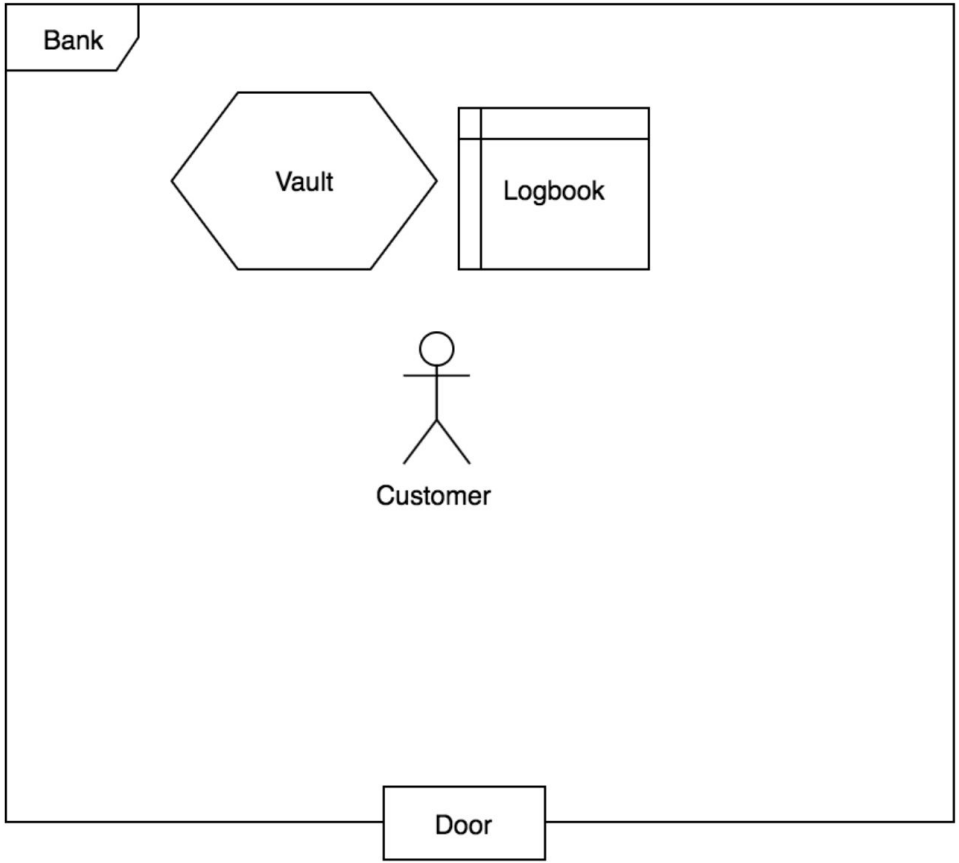
Application Programming Interface (API): a system made to help us use some software in an automated way

Web-scraping vs APIs

Web Scraping / Crawling	APIs
Work Intensive	Easily Automated
Subject to Change	Consistent / Predictable
Referred to as ‘Soup’	Intentionally Structured
Generally Inefficient	Efficient

Imagine you own a bank

- Vault for storing money,
- Logbook for recording how much money people have
- You run bank by granting open access to everybody
- No checks are carried out, and you trust people to do the right thing



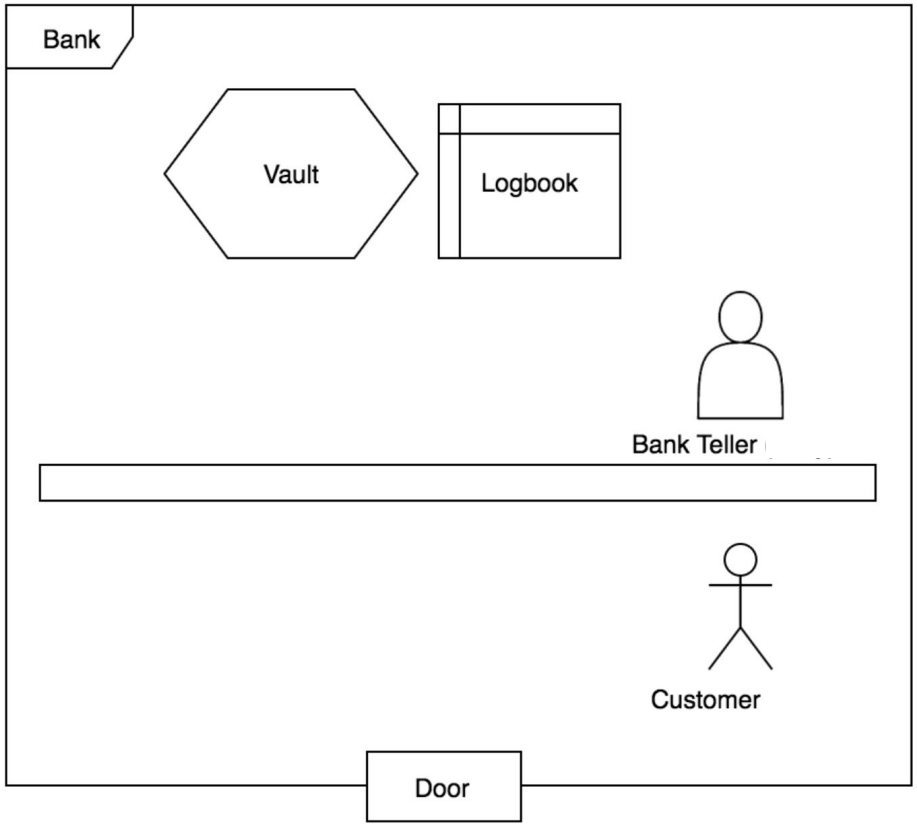
Customer wants to deposit Money

- Go to the bank with the money in their hands
- Open the vault and put the money inside
- Go to the logbook and write “I am <name> and I deposited <amount> into the vault
- Leave the bank

What’s the problem with that approach?

Let’s Hire Someone

- We have a wall separating the customer from the logbook and the vault
- If I want to deposit or withdraw money I have to speak to the person we hired



Customer wants to deposit Money

- Go to the bank with the money in their hands
- Approach our employee and say: I would like to deposit <amount> into my account
- Employee takes the money, opens the vault and puts the money in
- Employee records this transaction on the log book
- Employee goes back to the customer, and tells them that the money has been deposited
- The customer leaves the bank

API - Application Programming Interface

Using APIs entails interacting with a system made to help us use some software in an automated way.

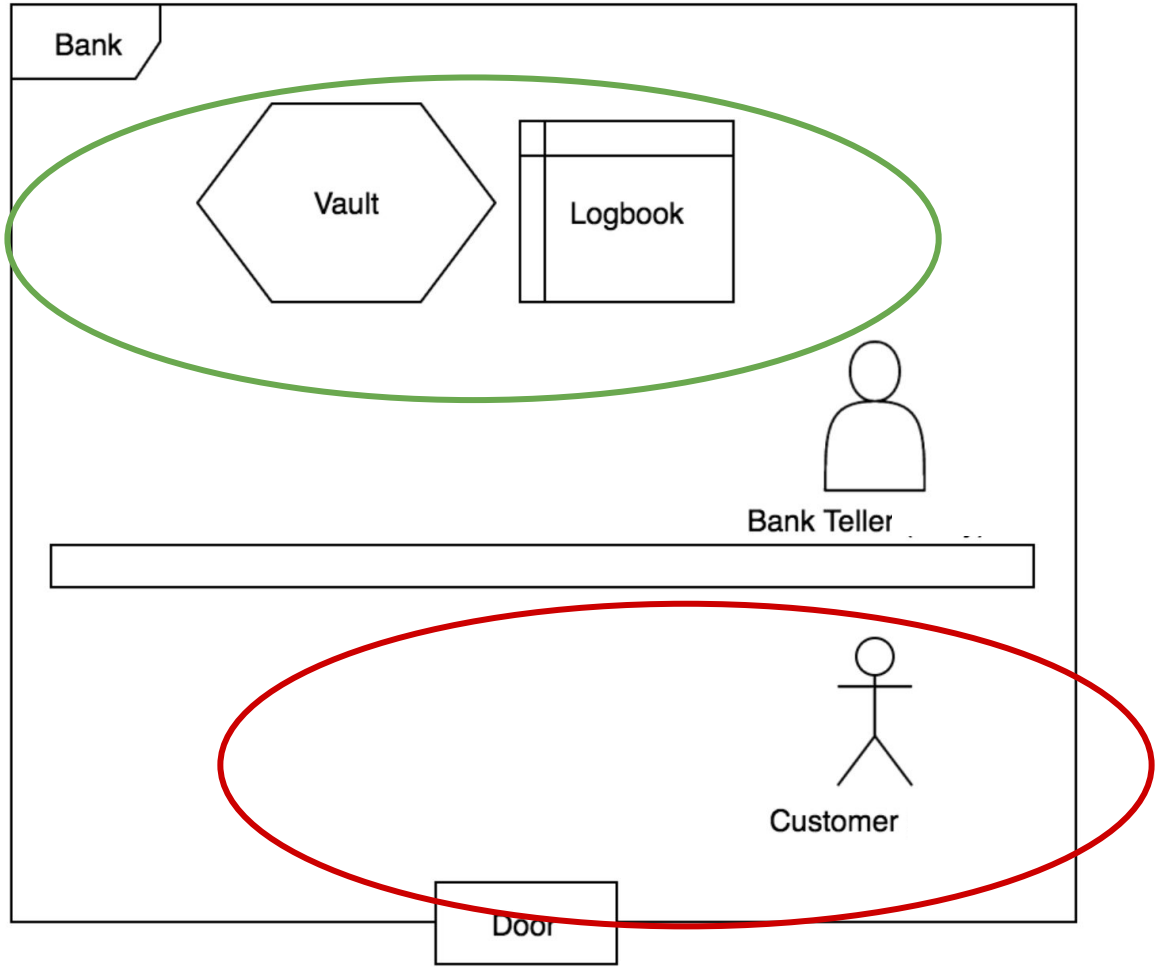
It's a set of clearly defined methods of communication between various software components

We can use APIs to (1) obtain data & (2) use tools.

An agreement that if you ask for information in a specified way the system will return information in a specific structure.

Interface

- The front area: customers queue up
- Back area: money handling happens
- The **middle** is where the front meets the back and exchange happens



Defined Communication: Protocol

- I want to withdraw money: I go to the bank counter, and say can I withdraw $500 from my account please?
- Our employee says: Yes, one moment please.
- The employee knows very well how to withdraw money and comes back to the counter, and says: Here’s your money. Goodbye
- Then I realize I haven’t done my Python assignment and ask the employee for help
- Employee does not know what to respond

A Protocol is a set of rules defining how components can interact with each other

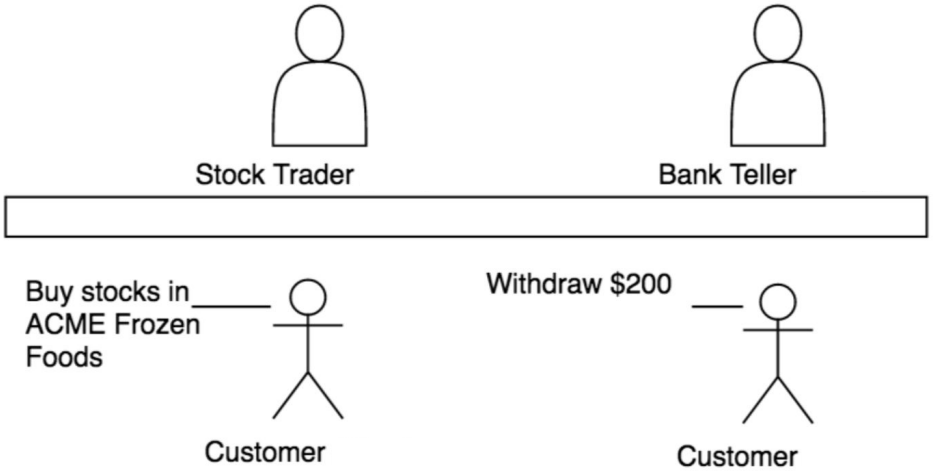
Format

- In our example the language in which the exchange happened was English
- The format specifies how you can encode the data you want to send over to the other party
- Both parties must understand and uphold the format
- Common formats for web APIs include XML and JSON

A Protocol is a set of rules defining how components can interact with each other

Endpoints

- We want to expand our business into stock markets as well
- We need a special kind of bank teller to deal with stock market trading.



A service provider that provides a specific subset of functions within the same Interface

What else can APIs do

- APIs can have verification logic built into it to ensure all operations are legal
- APIs can have error reporting mechanisms to indicate an error has occurred
- Pagination/Filtering
- Authorization and access control
- Implement rate-limiting to control the use of server resources, to ensure users cannot abuse the service

What is JSON

JavaScript Object Notation, and is a way to store information in an organized, easy-to-access manner.

```
var dani = {
      "age" : "38",
      "hometown" : "New York, NY",
      "sex" : "female"
};
```

Object we access using the variable **dani**. By enclosing the variable's value in curly braces, we're indicating that the value is an object. Inside the object, we can declare any number of properties using a "name": "value" pairing, separated by commas.

JSON continued

To access the information stored in dani, we point to the name of the property we need.

```
document.write('Dani is ' dani.age); // Output: Dani is 38
document.write('Dani is ' dani.sex); // Output: Dani is female
```

JSON continued - Arrays

```
var family = [{
  "name" : "Dani",
  "age" : "38",
  "sex" : "female"
},
{
  "name" : "Steffi",
  "age" : "35",
  "sex" : "female"
}];
```

We can access each element

```
document.write(family[1].name); // Output: Steffi
document.write(family[0].age); // Output: 38
```