API Protocols

Protocols

- defines how the API connects to the internet and how it communicates information
- determines how to design and build API as well as what's required to maintain it

Types of API Protocols

1. REST API (Representational State Transfer)
   - defines routes with URL
   - advantage
     - require the developer to wrap routes with XML
   - disadvantage
     - can only transmit information through the HTTP protocol
       - limited to sending text
   - architectural requirements
     - client-server architecture
     - statelessness
     - cacheable
     - layered
2. SOAP API (Simple Object Access Protocol)
   - can communicate over other internet communication protocols (TCP, SMTP, HTTP)
   - advantage
     - more flexible than REST
   - disadvantage
     - can only work with XML data
     - strict requirements for requests
3. RPC API (Remote Procedure Call)
   - can return XML or JSON responses
   - calls a method rather than a data resource
   - the response from the server is confirmation that the function was triggered/failed
   - used for distributed client-server applications

URI (Universal Resource Identifier)

- identifies where a specific resource can be found; such as a page or document

Using MongoDB in Flask Application

PyMongo

- a MongoDB database driver that allows interact with the MongoDB database in python

Connecting to MongoDB Library

- MongoClient creates a client object for a MongoDB instance called client, which allows to connect and interact with the MongoDB server

Sample Code:

```python
from flask import Flask
from pymongo import MongoClient

app = Flask(__name__)

client = MongoClient('localhost', 27017)

db = client.flask_db
```