


# REST



in 2000 **Roy Fielding** introduced REST in  
PhD dissertation



```
fetch('https://myapi.com/v1/users', {  
  method: 'POST',  
  headers: {  
    'Content-Type': 'application/json',  
    'Accept': 'application/json',  
    'Authorization': 'Bearer YOUR_SECRET_TOKEN_HERE'  
  }  
  body: {  
    name: 'Jane Doe',  
    email: 'jane@example.com'  
  }  
})
```



```
<definitions name="BankService">
```

```
<types>
```

```
  <element name="GetAccountBalanceRequest">
```

```
    <complexType>
```

```
      <sequence>
```

```
        <element name="AccountId" type="string"/>
```

```
      </sequence>
```

```
    </complexType>
```

```
  </element>
```

```
  <element name="GetAccountBalanceResponse">
```

```
    <complexType>
```

```
      <sequence>
```

```
        <element name="Balance" type="decimal"/>
```

```
        <element name="Currency" type="string"/>
```

```
      </sequence>
```

```
    </complexType>
```

```
  </element>
```

```
</types>
```

```
<portType name="BankPort">
```

```
  <operation name="GetAccountBalance"/>
```

```
</portType>
```

```
</definitions>
```

**what data type accept?**

**What responses look?**

**What Operations exist?**



traditional api model

# GraphQL



**Instead of the server** deciding what data you get, **the client** tells the server exactly what it wants



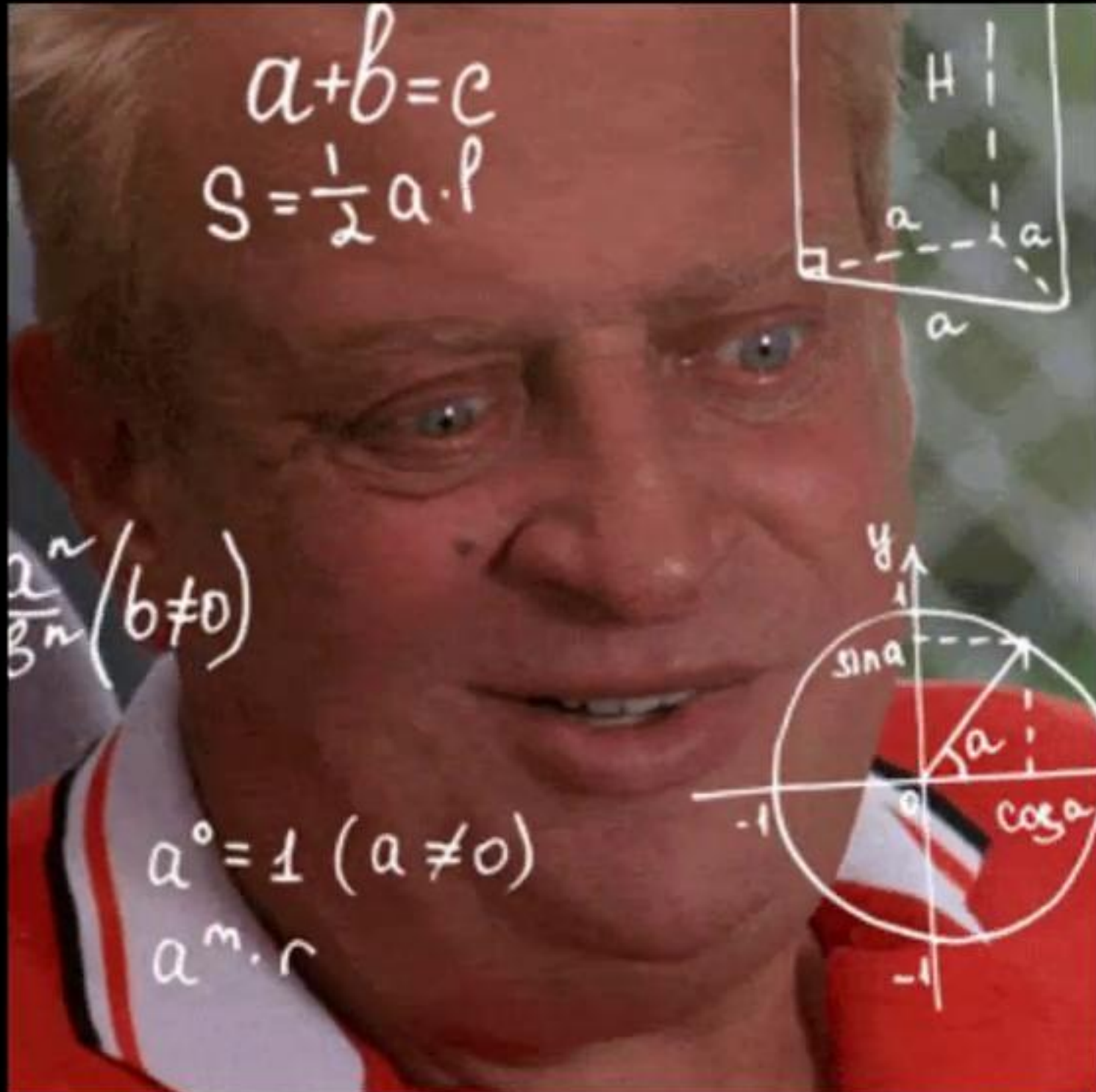


**Server must ...**

**limit queries**

**enforce depth(field) limits**

**manage overloading**



# gRPC



A curved arrow labeled "Uses" points from the text "gRPC" down to the text "Binary protocol over HTTP/2".

✗ Not plain HTTP + JSON

✓ **Binary protocol** over **HTTP/2**

```
const express = require("express");
const app = express();

app.use(express.json());

app.post("/webhook/payment", (req, res) => {


  const event = req.body;

  if (event.type === "payment.success") {
    console.log("Payment confirmed");
  }

  res.sendStatus(200);

});

app.listen(3000);
```



**sends an HTTP request**



```
import { WebSocketServer } from 'ws';

const wss = new WebSocketServer({ port: 8080 });

wss.on('connection', (socket) => {

  socket.on('message', (message) => {
    console.log('Received:', message.toString());

    socket.send('Message received! 🖐️');

    socket.on('close', () => {
      console.log('Client disconnected ❌');
    });
  });
});
```

**Message from client**



**Server talks back**





# Webhooks

solved the “please stop polling” problem

```
pcibios_init : BIOS32 Service Directory structure at 0x00000000
pcibios_init : BIOS32 Service Directory entry at 0x00000000
pcibios_init : PCI BIOS revision 2.00 entry at 0x00000000
Probing PCI hardware.
Calibrating delay loop.. ok - 36.84 BogoMIPS
Memory: 14984k/16384k available (552k kernel code, 384k reserved
NET3.035 for Linux 2.0
NET3: Unix domain sockets 0.13 for Linux NET3.035.
TCP/IP for NET3.034 IP Protocols: ICMP, UDP, TCP
VFS: Diskquotas version dqquot_5.6.0 initialized
Checking 386/387 coupling... Ok, fpu using exception 16 error re
Checking 'hlt' instruction... Ok.
Intel Pentium with F0 0F bug - workaround enabled.
alias mapping IDT readonly ... .. done
Linux version 2.0.35 (root@rabbit) (gcc version egcs-2.90.29\
980515 (egcs-1.0.3 release)) #3 Fri Nov 13 15:07:45 CET 1998
Starting ksvapd v 1.4.2.2 Serial driver version 4.13 with no ser
tty00 at 0x03f8 (irq = 4) is a 16550A
tty01 at 0x02f8 (irq = 3) is a 16550A
APM BIOS not found.
Real Time Clock Driver v1.09
Configuring Adaptec (SCSI-ID 7) at IO:330, IRQ 11, DMA priority
scsi0 : Adaptec 1542
scsi : 1 host
```