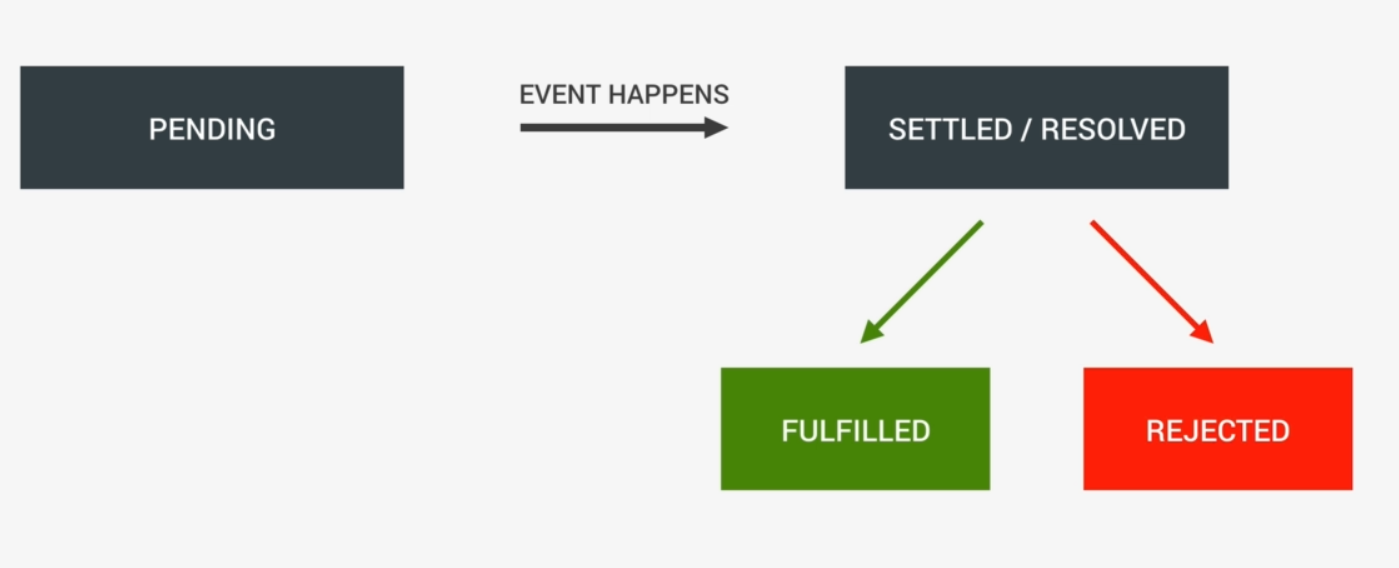


Promise như một lời hứa

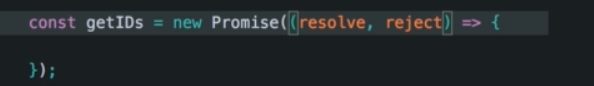
Ví dụ: lấy dữ liệu từ server -> promise hứa rằng dữ liệu sẽ được lấy trong tương lai -> lấy dữ liệu



We can produce or consume promises

Produce: create a new promise and send a result using that promise

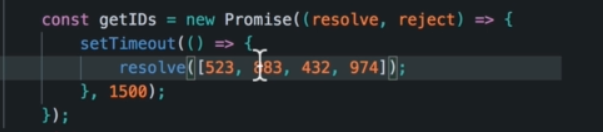
Consume: use callback function for fulfillement or rejection



Promise successful -> call resolve function

Promise fail -> call reject function

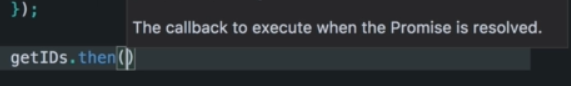
produce



Không cần thiết gọi hàm rejecte vì setTimeout ko fail

ở hàm trên -> khi thành công -> mãng trả về cho Promise -> getIDs chứa mãng

cunsume(chưa chắc)

(fullfiled)

getIDs.then() : thực hiện cái hàm callback trong đó khi promised successful

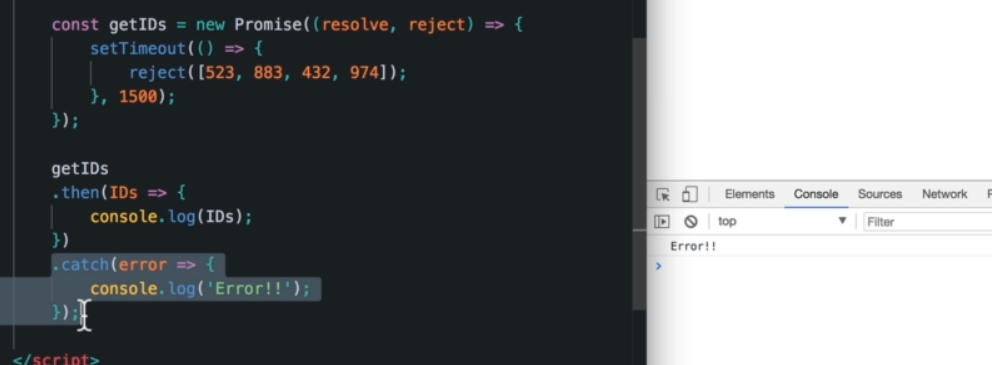


Ids ở đây là giá trị của resolve (kết quả của sucessful promise)

* Trong callback của getIDs.then() thực hiện những việc sau khi promise thành công



Catch giống then nhưng là trường hợp fail



Hàm callback trong catch hoặc then được gọi ngay sau khi nhận được kết quả từ promise

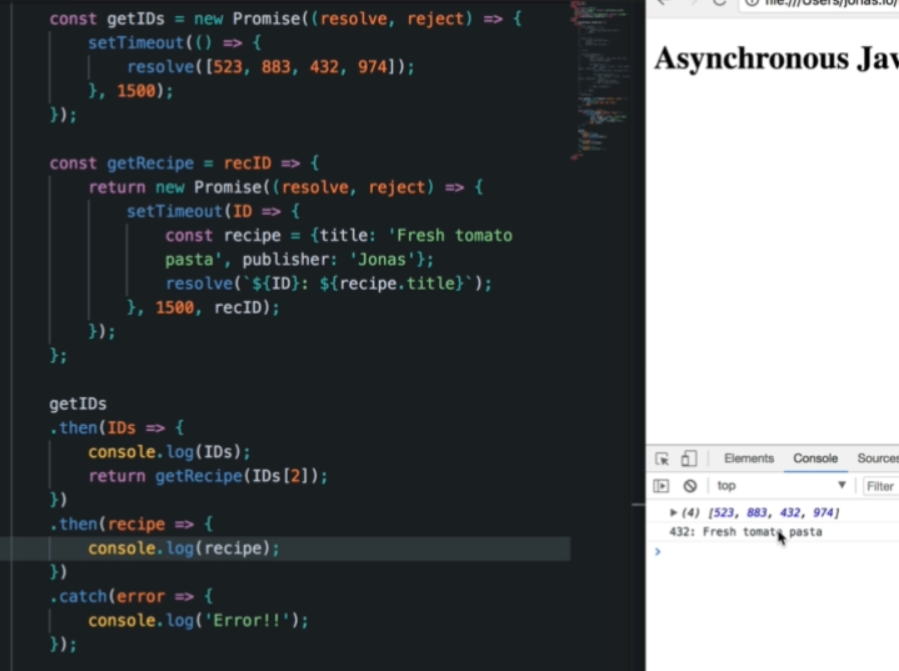
// chồng lớp ( giải quyết callback hell trong phần trước)



Sử dùng hàm với biến để truyền biến recID lấy được từ vòng trên

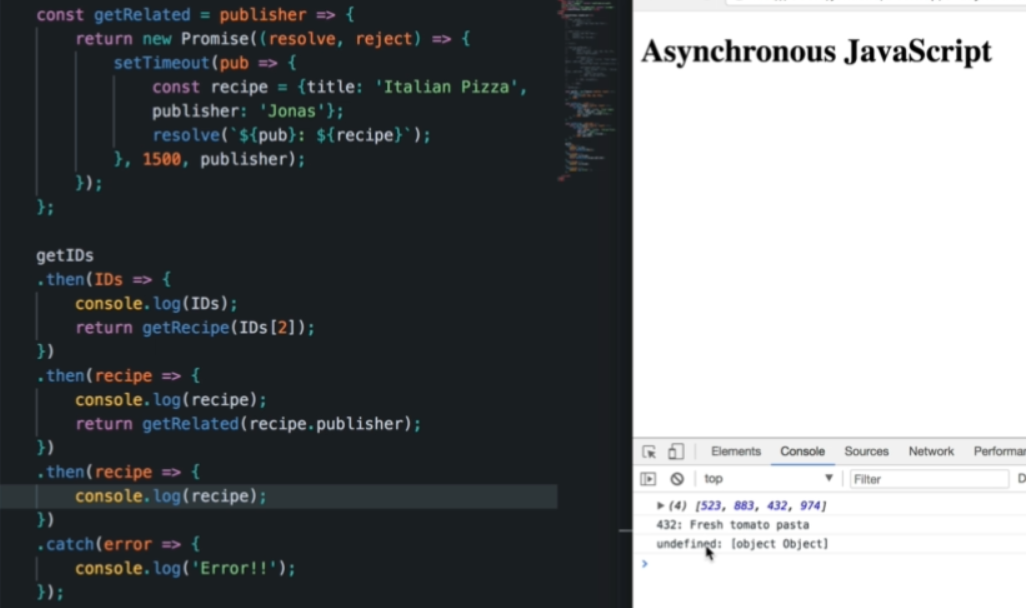
Biến recID sẽ được truyền cho hàm setTimeout

* Consume: bằng cách gọi hàm rescipe(ids[20]).then trong then của vòng trước -> hell -> không giải quết được vấn đề



Để tránh cái này thì retturn cái id cần lấy rồi .then sau đó, tương tự với catch

Vòng 3



Undefin ở đây do trong ví dụ ko phải object -> không quan trọng, khi làm thực tế sẽ đúng