**Javascript Promises and async / await**

**Promises**

A promise in javascript is a object that represents the eventual completion of asynchronous operation and its resulting value. Promise provide a way to handle asynchronous code in a more readable and maintainable manner compared to traditional callback function

A promise object has a state that can be

* **Pending**

The initial state. The operation hasnt completed yet

* **Fulfilled**

The operation completed successfully, and the promise has a resolved value.

* **Rejected**

The operation failed, and the promise has a reason for the failure.

Promise chaining

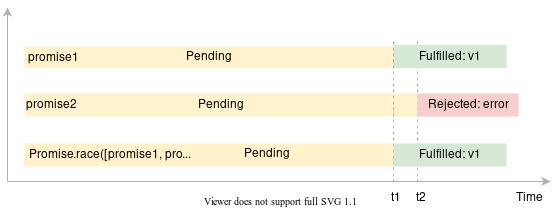
Promise.all()

The promise.all() method accepts a list of promises and returns a new promise that resolves to array of results of the input promises if all the input promises are resolved / rejected with error of the first rejected promise.

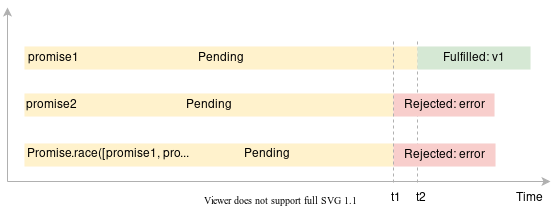
Promise.all() method is best for aggregate results from multiple asynchronous operations.

Promise.race()

The promise.race(iterable) method returns a new promise that fulfills / rejects as soon as one of the promises in iterable fulfills / rejects, with the value / error from that promise.



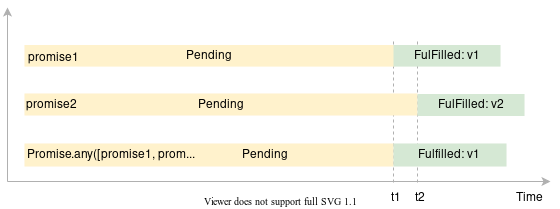
* The promise1 is fulfilled with the value v1 at t1.
* The promise2 is rejected with the error at t2.
* Because the promise1 is resolved earlier than the promise2, the promise1 wins the race. Therefore, the Promise.race([promise1, promise2]) returns a new promise that is fulfilled with the value v1 at t1



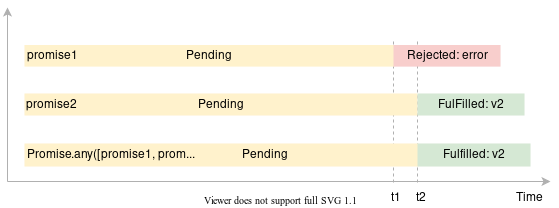
* The promise1 is fulfilled with v1 at t2.
* The promise2 is rejected with error at t1.
* Because the promise2 is resolved earlier than the promise1, the promise2 wins the race. Therefore, the Promise.race([promise1, promise2]) returns a new promise that is rejected with the error at t1

Promise.any()

If one of the promises in the iterable object is fulfilled, the Promise.any(iterable) returns a single promise that resolves to a value.



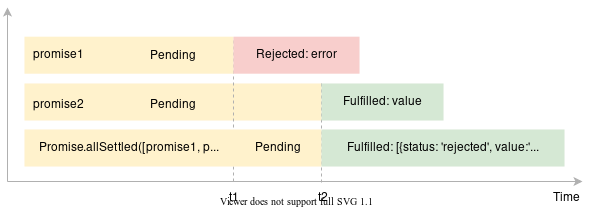
* The promise1 resolves to a value v1 at t1.
* The promise2 resolves to a value v2 at t2.
* The Promise.any() returns a promise that resolves to a value v1, which is the result of the promise1, at t1



* The promise1 is rejected with an error at t1.
* The promise2 is fulfilled to value v2 at t2.
* The Promise.any() returns the promise that resolves to a value v2 which is the result of the promise2. Note that the Promise.any() method ignores the rejected promise (promise1)

Promise.allSettled()

The promise.allSettled() methods accept iterable of promises and returns a new promise that resolves when every input promise has settled with array of object that describe the result of each promise.



* The promise1 rejects to the error at t1.
* The promise2 resolves to a value at t2.
* The Promise.allSettled() method resolves to an array containing objects that describe the statuses and outcomes of the promise1 and promise2

Promise.prototype.finally()

Promise error handling

async / await

Promise.withResolvers()

Error handling

try / catch

try / catch / finally

throw exception

optional catch binding