

promise

直列1

```
var app = new Vue({
  el: "#app",
  data: {
    greeting: "こんにちは",
  },
  created: function() {
    this.createPromise(1000)
      .then(() => console.log('成功1秒後'))
      .catch(() => console.log('失敗'))
    this.createPromise(2000)
      .then(() => console.log('成功2秒後'))
      .catch(() => console.log('失敗'))
    this.createPromise(3000)
      .then(() => console.log('成功3秒後'))
      .then(() => console.log('全て終了'))
      .catch(() => console.log('失敗'))
  },
  methods: {
    createPromise: function(time) {
      return new Promise((resolve, reject) => {
        setTimeout(() => {
          resolve();
          reject();
        }, time);
      })
    },
  },
});
```

直列2

```
var app =new Vue({
  el:"#app",
  data:{
    greeting:"こんにちは",
  },
  created:function(){
    this.createPromise(1000)
    .then(()=>console.log('成功1秒後'))
    .then(()=>{
      this.createPromise(2000)
      .then(()=>console.log('成功2秒後'))
    })
    .then(()=>{
      this.createPromise(3000)
      .then(()=>console.log('成功3秒後'))
      .then(()=>console.log('全て終了'))
    })
    .catch(()=>console.log('失敗'))
  },
  methods:{
    createPromise:function(time){
      return new Promise((resolve,reject)=>{
        setTimeout(()=>{
          resolve();
          reject();
        },time);
      })
    },
  },
});
```

並列1

```
var app =new Vue({
  el:"#app",
  data:{
    greeting:"こんにちは",
  },
  created:function(){
    console.log('aaaaa');

    Promise.all([
      this.createPromise1(),
      this.createPromise2(),
      this.createPromise3()
    ])
    .then((data)=>{
      console.log(data);
      console.log('全て終了');
    })
    .catch(()=>{
      console.log('失敗');
    })
  },
  methods:{
    createPromise1:function(){
      return new Promise((resolve,reject)=>{
        setTimeout(()=>{
          resolve('1つ目');
          reject();
        },1000);
      })
    },
    createPromise2:function(){
      return new Promise((resolve,reject)=>{
        setTimeout(()=>{
          resolve('2つ目');
          reject();
        },2000);
      })
    },
    createPromise3:function(){
      return new Promise((resolve,reject)=>{
        setTimeout(()=>{
          resolve('3つ目');
          reject();
        },5000);
      })
    },
  },
});
```

並列2(ただ単の並列)

```
var app =new Vue({
  el:"#app",
  data:{
    greeting:"こんにちは",
  },
  created:function(){
    console.log('aaaaa');

    this.createPromise1()
      .then(()=>console.log("1つ目"))
      .catch(()=>console.log('失敗'))

    this.createPromise2()
      .then(()=>console.log("2つ目"))
      .catch(()=>console.log('失敗'))
    this.createPromise3()
      .then(()=>console.log("3つ目"))
      .then(()=>console.log('全て終了'))
      .catch(()=>console.log('失敗'))
  },
  methods:{
    createPromise1:function(){
      return new Promise((resolve,reject)=>{
        setTimeout(()=>{
          resolve('1つ目');
          reject();
        },1000);
      })
    },
    createPromise2:function(){
      return new Promise((resolve,reject)=>{
        setTimeout(()=>{
          resolve('2つ目');
          reject();
        },2000);
      })
    },
    createPromise3:function(){
      return new Promise((resolve,reject)=>{
        setTimeout(()=>{
          resolve('3つ目');
          reject();
        },5000);
      })
    },
  },
});
```