



UI5con

# Handling of temporarily unavailable backends

How the OData V2 & V4 Models can be configured to retry requests which are responded with an HTTP 503 by the backend.



**Viktor Sperling**

Developer  
SAP SE



**Mathias Uhlmann**

Development Manager  
SAP SE

# Agenda



01

Introduction

02

Demo

03

How to

04

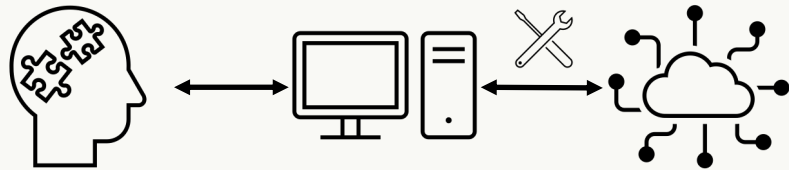
Q&A



# Introduction

# Introduction

## Motivation

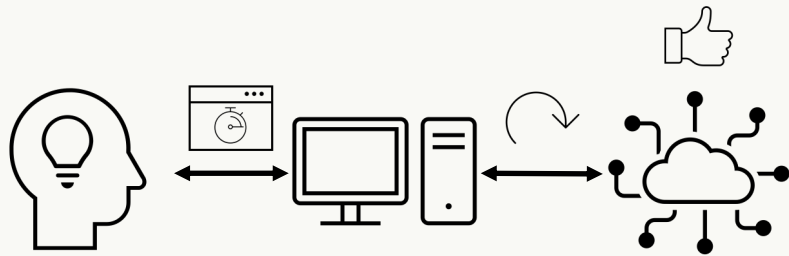


- All systems require maintenance activities
- Techniques like Zero Downtime do not eliminate all downtimes
- Keep the downtime short and let the system be temporarily not available
- All HTTP requests are replied to with status 503 and a retry-after header
- Very inconvenient for users during this short period of time if application breaks

# Introduction

## ODataModel functionality in a nutshell

- Allow to repeat the failed request after the communicated time period
- I.e. do not fail (yet) and do not report errors (to the MessageModel)
- Provide a mean to the application to inform the user and control the behavior
- Done by Fiori elements V2 and Fiori elements V4





# Demo



# How to

What do I need to do?

# The API

## Model.setRetryAfterHandler

- Available on all OData Models (V2 and V4)
- Supports all requests after the Models initialisation
- Registers the given call back function which is called in case of an HTTP 503 error
  - The function has a parameter containing relevant data for the retry, `{message: string, retryAfter: Date}`
  - See also, [sap/ui/model/odata/v2/RetryAfterError](https://api.sap.com/docs/api/odata/v2/RetryAfterError) for V2 Models
  - In case of V4, the call back receives an actual error instance with these two properties



```
this.getView().getModel().setRetryAfterHandler((oRetryAfterError) => {  
    // prepare stuff  
    return new Promise((fnResolve, fnReject) => {  
        // do stuff while backend is unavailable  
    });  
});
```



# The prerequisites

- The backend has to return a retry-after header in case of 503
- The **Model.setRetryAfterHandler** call back function has to return a Promise
- The Promise has to be resolved/rejected eventually in the function
- During the process the UI has to be blocked

```
this.getView().getModel().setRetryAfterHandler((oRetryAfterError) => {  
    // prepare stuff, e.g. calculate seconds until the retry  
    return new Promise((fnResolve, fnReject) => {  
        // do stuff while backend is unavailable  
        setTimeout(() => {  
            // resolve after the calculated time  
            fnResolve();  
        }, iRetryAfterSeconds * 1000);  
    });  
});
```

# Recommendations

## General recommendations

- Keep the handling of 503-Errors consistent for all Models in your application
- All Models should use the same Promise and the same Retry-After-Handler

## Possible rules regarding the waiting

- if (`oRetryAfterError.retryAfter`  $\leq$  5 seconds)
  - $\Rightarrow$  lock the screen and automatically retry after given time
- if (5 seconds  $<$  `oRetryAfterError.retryAfter`  $\leq$  10 minutes)
  - $\Rightarrow$  show dialog, wait and offer a reject button
- if (10 minutes  $<$  `oRetryAfterError.retryAfter`)
  - $\Rightarrow$  do not show dialog, do not wait, reject Promise immediately





# Expert Corner "UI5 Models"



13:25 – 14:45



# Q&A



# Got feedback? Let's hear it!



Handling of temporarily  
unavailable Backends

The image shows a smartphone screen with a feedback form titled "Keynote". The form has a yellow header bar with a back arrow on the left and a star icon on the right. Below the header, the title "Keynote" is centered. The main content area is white and contains two sections of feedback questions. The first section asks "Overall, how satisfied were you with this session? (5=Very Satisfied)" and has five radio button options: "5 - Very satisfied", "4 - Satisfied", "3 - Neutral", "2 - Dissatisfied", and "1 - Very dissatisfied". The second section asks "How relevant was the session content to your work? (5=Very Relevant)" and has five radio button options: "5 - Very satisfied", "4 - Satisfied", "3 - Neutral", "2 - Dissatisfied", and "1 - Very dissatisfied".

Keynote

Feedback

Overall, how satisfied were you with this session? (5=Very Satisfied)

☐ 5 - Very satisfied

☐ 4 - Satisfied ☐ 3 - Neutral

☐ 2 - Dissatisfied

☐ 1 - Very dissatisfied

How relevant was the session content to your work? (5=Very Relevant)

☐ 5 - Very satisfied

☐ 4 - Satisfied ☐ 3 - Neutral

☐ 2 - Dissatisfied

☐ 1 - Very dissatisfied