|  |  |  |
| --- | --- | --- |
|  | Development of ITCM Export Tool with SAP Fiori – documentation of changes | SAP Field Services |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Proposed Approach** | | |
| **History** | |  | | |
| Version | Author | | Status | Date |
| 1.0 | Gayathri Vijayakumar | | Initial Version | 2022-09-17 |

## General Purpose and Scope

|  |
| --- |
| * 1. Requirement |
| The idea behind the application to make the relevant representation of well-known ABAP reports and should have the same functions as the relevant transaction on backend /ITCM/ET\_BASKET\_UNDO using FIORI UI5 on BTP platform.   The idea is to get together all the apps which can be easily accessible using FIORI Launchpad "ITCM Export Tool" to the category "ITC - International Trade Control - Export Management". |

## 2. SAP Fiori File Description

|  |
| --- |
| 2.1 Table of Content |
| |  |  |  |  | | --- | --- | --- | --- | | S.No. | Controller Name | Methods | Description | | 1. | Initial.controller.js | [onReqrowselection](#_onReqrowselection) | It helps to navigate to detail page and pass selected ‘Request ID’ as a parameter. | | 2. | Initial.controller.js | <onNav_reqlist> | It is used to navigate to detail page. | | 3. | Detail.controller.js | <onInit> | Used to trigger attachPatternmatched() | | 4. | Detail.controller.js | \_[routePatternMatched](_routePatternMatched) | It checks if the pattern is matched and binds model data. | |

|  |
| --- |
| 2.2 neo-app.json |
|  |

|  |
| --- |
| 2.3 Manifest.json |
|  |

|  |
| --- |
| 2.4 Initial.view.xml |
|  |

|  |
| --- |
| 2.5 DETAIL.view.xml |
|  |

|  |
| --- |
| 2.6 Initial.controller.js |
| [onReqrowselection](#_onReqrowselection) – Used to navigate and pass ‘RequestId’ as a parameter to a detail page.    <onNav_reqlist> – Navigates from initial screen and follows a pattern defined in Manifest.json |

|  |
| --- |
| 2.7 detail.controller.js |
| <onInit>– Used to check if the route pattern is matched.    \_[routePatternMatched](_routePatternMatched) – |

|  |
| --- |
| 2.8 Component.js |
|  |

|  |
| --- |
| 2.9 Mockserver.html |
|  |

|  |
| --- |
| 2.10 Mockserver.js |
|  |

|  |
| --- |
| 3.1 Mockserver |
| To start application with mock server data, developer need to:  1) Right click project in the Webide,  2) select Run -> Run Configurations,  3) select Run as Web Application,  4) click plus button to create no run template,  5) give a name for a configuration,  6) set Run Application File to /webapp/test/mockserver.html,  7) select Run with mock data and click Save and Run button. Later, user may start app by simply selecting Run -> Run (name of created template) |

|  |
| --- |
| 3.2 UI Screens |
|  |

## 4. Related Changes

|  |
| --- |
| 4.1 Requests |
| In order to access the application code via GIT repository and perform the modification the user needs to have relevant authorization:  (See for example ARM request 4551391)  - ApplicationUserAdministrator (SPD\_IDD)  - Developer (SPD\_IDD)  - WebIdeDeveloper (SPD\_IDD) |