

CAMUNDA COMMUNITY SUMMIT

2021

Session Notes:

<Session 4>

Topic: Best Practices for Integrating a Custom Frontend with Camunda Platform

Moderator: Martin Stamm

Interesting Notes:

FrontEnd

- Get Data + Display Data
- Modify Data
- BPMN diagrams display + interaction

FE app -> workflow engine

BPMN-js as a leveraging library for app development (see <https://bpmn.io/toolkit/bpmn-js/>)

Best Practices for building a custom Tasklist: <https://camunda.com/best-practices/building-a-custom-tasklist-application/>
<https://camunda.com/best-practices/enhancing-tasklists-with-business-data/>

Form builder libraries

<https://forms.io>

<https://uniforms.tools>

<https://github.com/bpmn-io/form-js>

<https://github.com/Taskana/taskana> (Open Source Task Management library)

Easier way to retrieve all data needed to present a form (NOT in 5 rest calls) -> GraphQL?

→ GraphQL community extension: <https://github.com/camunda/camunda-bpm-graphql>

-> Camunda spike ticket to prototype a new task query API:

<https://jira.camunda.com/browse/CAM-13160> (ideas for technologies welcome, just comment on the ticket :)

Websocket to keep track of tasks

CAMUNDA COMMUNITY SUMMIT

2021

-> Camunda ticket to implement listeners as external tasks so that you can remove “technical” external tasks for notifications from your process: <https://jira.camunda.com/browse/CAM-13197>

High-Level Takeaways:

- 1) Leverage BPMN-js
- 2) Extend the REST-API to fit your needs
- 3) Use form-key as key for external form querying (see <https://docs.camunda.org/manual/latest/reference/bpmn20/tasks/user-task/#retrieving-the-form-key-using-the-form-service> or via REST <https://docs.camunda.org/manual/latest/reference/rest/task/get-form-key/>)