OZG-Cloud | Technical Documentation | Application Room

Summary

The application room MVP acts as an interface between applicants and caseworkers and serves as a sort of mailbox for the applicants. The application room allows applicants to respond to messages from caseworkers by easily uploading requested, relevant documents.

The application room will undergo further development after we have established the MVP version of it. Therefore, we need to enable other individuals to contribute to the development of the microservice.

We aim to support this effort by creating comprehensive technical documentation. Some points to consider:

- Viewpoint: "If I would be (as a developer) new to this area in the code, what would I need to dig into it?"
- Include the solution steps if they are not so obvious
- Document needed migration scripts

Content

Motivation

Overview

Ticket & Implementation

Ticket(s):

mgm-Expert:

Configuration

IntelliJ Settings

Because the code will be shipped as Open Source we need to add proper license headers to the souce files. In the OZG-Cloud Project the EUPL License version 1.2 is used therefore we use the same license.

To add the license header the IntelliJ Copyright plugin is used.You'll find a guide how to use it here

You can import the Copyright Profile using the Eupl1_2.xml file and the setting by us the profile_settings.xml file

Eupl1_2.xml

profiles_settings.xml

As described in the guide the the "Save Actions" in IntellJ must be enabled to automatically add and update the license header.

Specification

Technical analysis

Implementation

Frontend

The front end uses react. For development node version 18 is used.

The Repository is ozg-frontend

Application Room

The microservice is implemented using Spring Boot.

Java Version 17 is used.

The Repository is ozg-application-room

Info Manager

The microservice is implemented using Spring Boot.

Java Version 17 is used.

The Repository ist ozg-info-manager

Further Details....

Frontend

Available Scripts

```
npm start
```

Runs the app in the development mode. Open http://localhost:8080 to view it in the browser. The page will reload if you make edits. You will also see any lint errors in the console.

```
npm run build
```

Runs all the build scripts

```
npm run build:typescript
```

Compiles the TypeScript files into JavaScript and stores the result in the build folder.

```
npm run test:unit
```

Launches the test runner in the interactive watch mode.
See the section about running tests for more information.

```
npm run test:e2e
```

To open the Cypress to create and run e2e test.

```
npm run check
```

Runs all the checks

```
npm run check:prettier
```

To run prettier find format problems with your JavaScript code.

```
npm run check:lint
```

To run eslint find and fix problems with your JavaScript code.

```
npm run prettier:write
```

To run prettier to format the sources.

Application Room

Building the service

Gradle is used as building system. Therefore to build the service run

```
./gradlew build
```

in the root directory of the project.

Starting the service

To start the service you can use this methods:

For Developers:

use gradle by running the command

```
    ./gradlew bootRun --args='--spring.profiles.active=standalone' for running a standalone version of the
Antragsraum with all external GRPC Services mocked
```

```
     ./gradlew bootRun --args='--spring.profiles.active=dev' for running a version of the Antragsraum where
you need the Informationmanager and a MongoDB
```

or by using IntelliJ Spring Tool and start the ApplicationRoomApplication in the Services window.

For Production:

to start the created artefact run

```
java -jar build/libs/ApplicationRoom-0.0.1-SNAPSHOT.jar
```

in the root directoy of the project

For further information see https://bitbucket.mgm-tp.com/projects/BUP/repos/ozg-application-room/browse/README.md

Info Manager

Building the service

Gradle is used as building system. Therefore to build the service run

```
./gradlew build
```

in the root directory of the project.

Starting the service

To start the service you can use this methods:

For Developers:

use gradle by running the command

```
./gradlew bootRun
```

or by using IntelliJ Spring Tool and start the InformationServiceApplication in the Services window.

For Production:

to start the created artefact run

```
java -jar build/libs/InformationService-0.0.1-SNAPSHOT.jar
```

in the root directoy of the project

For even further information see https://bitbucket.mgm-tp.com/projects/BUP/repos/ozg-info-manager/browse/README.md

Interface Descriptions

Application Room

Draft

The following Interface documentation is generated from the Interface description files (protobuf files) of the pluto-interface lib Version 1.6.0-SNAPSHOT. These files are created by the OZG-Cloud Team and can be changed on the next versions.

All services to the OZG Cloud are mocked right now until the the services are finalized on OZG-Cloud side. There is no connection so far.

binaryfile.proto

GrpcBinaryFile

Field	Type	Label	Description
id	string		
name	string		
size	int64		
contentType	string		

GrpcBinaryFileDataRequest

Field	Type	Label	Description
context	GrpcCallContext		
fileId	string		

GrpcGetBinaryFileDataRequest

Field	Type	Label	Description
context	GrpcCallContext		
fileId	string		

GrpcGetBinaryFileDataResponse

Field	Type	Label	Description
fileContent	bytes		

GrpcUploadBinaryFileMetaData

Field	Type	Label	Description
context	GrpcCallContext		

vorgangId	string		
field	string		
fileName	string		
contentType	string		
size	int64		

GrpcUploadBinaryFileRequest

Field	Type	Label	Description
metadata	GrpcUploadBinaryFileMetaData		
fileContent	bytes		

GrpcUploadBinaryFileResponse

Field	Type	Label	Description
fileId	string		

BinaryFileService

Method Name	Request Type	Response Type	Description
UploadBinaryFileAsStream	GrpcUploadBinaryFileRequest stream	GrpcUploadBinaryFileResponse	
GetBinaryFileContent	GrpcGetBinaryFileDataRequest	GrpcGetBinaryFileDataResponse stream	
FindBinaryFilesMetaData	GrpcBinaryFilesRequest	GrpcFindFilesResponse	
GetAttachments	GrpcGetAttachmentsRequest	GrpcGetAttachmentsResponse	
GetRepresentations	GrpcGetRepresentationsRequest	GrpcGetRepresentationsResponse	

Top

postfach.model.proto

GrpcFindRueckfragenRequest

Field	Type	Label	Description
postfachId	string		

GrpcGetRueckfragRequest

Field	Type	Label	Description
rueckfrageId	string		

GrpcFindRueckfragenResponse

Field	Type	Label	Description
rueckfrage	GrpcRueckfrage	repeated	List of rueckfragen

GrpcGetRueckfragenResponse

Field	Type	Label	Description
rueckfrage	GrpcRueckfrage		

GrpcRueckfrage

Field	Type	Label	Description

id	string		
vorgangId	string		
vorgangName	string		
sentAt	string		
answeredAt	string		
status	string		
text	string		
answers	GrpcRueckfrageAnswer	repeated	

GrpcRueckfrageAnswerRequest

Field	Type	Label	Description
answer	GrpcRueckfrageAnswer		

GrpcRueckfrageAnswer

Field	Type	Label	Description
rueckfrageId	string		
answerText	string		
answerAttachmentFileId	string	repeated	

PostfachService

Method Name	Request Type	Response Type	Description
FindRueckfragen	GrpcFindRueckfragenRequest	GrpcFindRueckfragenResponse	Gets all rueckfragen of a Postfach
GetRueckfrage	GrpcGetRueckfrageRequest	GrpcGetRueckfrageResponse	Get a rueckfrage by its id
SendRueckfrageAnswer	GrpcRueckfrageAnswerRequest	GrpcCommand	

GrpcCommand

Field	Type	Label	Description
id	string		
vorgangId	string		
createdAt	string		
finishedAt	string		
createdBy	string		
createdByName	string		
status	string		
errorMessage	string		
relationId	string		
orderString	string		

GrpcCallContext

Field	Type	Label	Description
client	string		
user	GrpcUser		
requestId	string		

GrpcUser

Field	Type	Label	Description
id	string		
name	string		

Info-Manager

This is the interface used between the Info-manager and the Antragsraum backend.

The interface is defined by the proto files you find at https://bitbucket.mgm-tp.com/projects/BUP/repos/ozg-info-manager/browse/ozg-info-manager-interface/src/main/protobuf/infomanager

Table of Contents

Top

infomanager/information.model.proto

GrpcInformationRequest

Field	Type	Label	Description
postfachId	string		The id of the users mailbox. In case of the BayernId this is the 'postfachId' of the BayernId service account

GrpcInformationResponse

Field	Type	Label	Description
nachrichten	GrpcNachricht	repeated	

GrpcServiceUrlRequest

Field	Type	Label	Description
nachrichtId	string		

GrpcServiceUrlResponse

Field	Type	Label	Description
url	string		

Top

infomanager/information.proto

InformationService

Method Name	Request Type	Response Type	Description
GetInformation	GrpcInformationRequest	GrpcInformationResponse	The service interface for requesting information of Nachrichten that belong to a post box

GetServiceUrlOfNachricht	GrpcServiceUrlRequest	GrpcServiceUrlResponse	Service interface to load the address of the origin of the event.

Nachrichten-Manager

> ⚠ **Draft**
>
> This interface definition is a draft of a possible interface between the Info-Manager and the NachrichtenManager in the OZG-Cloud

This is the interface used between the Info-manager and the NachrichtenManager in the OZG-Cloud.

The interface is defined by the proto files you find at https://bitbucket.mgm-tp.com/projects/BUP/repos/ozg-info-manager/browse/ozg-info-manager-interface/src/main/protobuf/nachrichteventmanager

Table of Contents

op

nachrichteventmanager/nachricht.model.proto

GrpcFinishedNachrichtRequest

Field	Type	Label	Description
nachrichtId	string		

GrpcNachricht

Field	Type	Label	Description
postfachId	string		Required The id of the users mailbox. In case of the BayernId this is the 'postfachId' of the BayernId service account
vorgangId	string		Required The id of the dossier created in the ozg-cloud.
nachrichtId	string		Required The id of the message send from the clerk to the customer using the ozg-cloud ALFA application
nachrichtenListUrl	string		Required The URL where the applicationroom backend can load the messages
rueckfrageId	string		Required The id of Rückfrage received from ozg-cloud

GrpcNewNachrichtRequest

Field	Type	Label	Description
nachricht	GrpcNachricht		

Top

nachrichteventmanager/nachricht.proto

NachrichtService

Method Name	Request Type	Response Type	Description
SaveNewNachricht	GrpcNewNachrichtRequest	.google.protobuf.Empty	Service interface for saving information about a new Nachricht
FinishNachricht	GrpcFinishedNachrichtRequest	.google.protobuf.Empty	Service interface removing information of Nachricht

Scalar Value Types

.proto Type	Notes	C++	Java	Python	Go	C#	PHP	Ruby
double		double	double	float	float64	double	float	Float
float		float	float	float	float32	float	float	Float
int32	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint32 instead.	int32	int	int	int32	int	integer	Bignum or Fixnum (as required)
int64	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint64 instead.	int64	long	int/long	int64	long	integer/string	Bignum
uint32	Uses variable-length encoding.	uint32	int	int/long	uint32	uint	integer	Bignum or Fixnum (as required)
uint64	Uses variable-length encoding.	uint64	long	int/long	uint64	ulong	integer/string	Bignum or Fixnum (as required)
sint32	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int32s.	int32	int	int	int32	int	integer	Bignum or Fixnum (as required)
sint64	Uses variable-length encoding. Signed int value. These more efficiently encode negative numbers than regular int64s.	int64	long	int/long	int64	long	integer/string	Bignum
fixed32	Always four bytes. More efficient than uint32 if values are often greater than 2^28.	uint32	int	int	uint32	uint	integer	Bignum or Fixnum (as required)
fixed64	Always eight bytes. More efficient than uint64 if values are often greater than 2^56.	uint64	long	int/long	uint64	ulong	integer/string	Bignum
sfixed32	Always four bytes.	int32	int	int	int32	int	integer	Bignum or Fixnum (as required)
sfixed64	Always eight bytes.	int64	long	int/long	int64	long	integer/string	Bignum
bool		bool	boolean	boolean	bool	bool	boolean	TrueClass/FalseClass
string	A string must always contain UTF-8 encoded or 7-bit ASCII text.	string	String	str/unicode	string	string	string	String (UTF-8)
bytes	May contain any arbitrary sequence of bytes.	string	ByteString	str	[]byte	ByteString	string	String (ASCII-8BIT)

How To's

How to extract the Postkorbhandle from log on dev

1. Login with BayernId on https://bup-ozg-ci-frontend.pidev.mgm-tp.com/
2. Open the Antragsraum App in Ranger https://tpirancher.mgm-tp.com/dashboard/c/c-m-ghlz9gdw/explorer/apps.deployment/bup-ozg-ci/ozg-antragsraum-server-deployment#pods
3. Select "View Logs" in the Popupmenu of the Antragsraum Pod (on the left, the three points)
4. Look for a line containing "CallContextServerInterceptor : Request Id - no request scope -; Set Auth token " and copy the line below. This is the Auth token
5. Open https://jwt.io/#debugger-io and past the Auth token. On the left side you'll see the decoded token and the value of "legacyPostkorbHandle' is the Postkorbhandle

Page History

Version	Date	Author	Comment
32	18.12.2023 10:35	Jens Reese	
31	15.12.2023 09:31	Jens Reese	
30	24.10.2023 15:06	Jens Reese	
29	24.10.2023 12:00	Jens Reese	
28	24.10.2023 11:28	Jens Reese	

Further Documentation

Title	Link
Design	
Technical	Interfaces/Configs - Injected Properties in Kubernetes Pods
Old	