MavPlus - Unofficial MÁV application Documentation

ON THIS PAGE

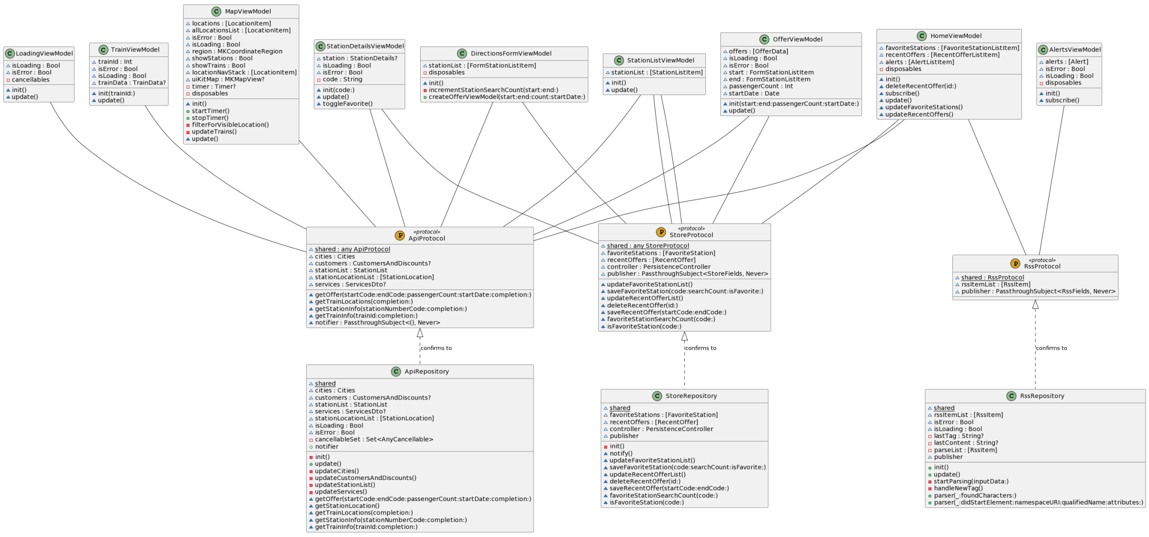
Summary

The application has been built with SwiftUI and the data it uses comes from the new MÁV Elvira API used by jegy.mav.hu

We used the MVVM Architecture with Repositories. There are 3 repositories in the architecture: ApiRepository (for initial and singular API calls), RSSRepository (for the news feed) and StoreRepository (to save and load user data to and from CoreData). There are protocols for each repository that should be implemented. These repositories are implemented as singleton classes.

Created by: Bálint Berente and Márton Pfemeter

Class diagram (view model and repository area)


LoadingViewModel
isLoading : Bool
isError : Bool
cancellables
init()
update()
TrainViewModel
trainId : Int
isError : Bool
isLoading : Bool
trainData : TrainData?
init(trainId:)
update()
MapViewModel
locations : [LocationItem]
allLocationsList : [LocationItem]
isError : Bool
isLoading : Bool
region : MKCoordinateRegion
showStations : Bool
showTrains : Bool
locationNavStack : [LocationItem]
uiKitMap : MKMapView?
timer : Timer?
disposables
init()
startTimer()
stopTimer()
filterForVisibleLocation()
updateTrains()
update()
StationDetailsViewModel
station : StationDetails?
isLoading : Bool
isError : Bool
code : String
init(code:)
update()
toggleFavorite()
DirectionsFormViewModel
stationList : [FormStationListItem]
disposables
init()
incrementStationSearchCount(start:end:)
createOfferViewModel(start:end:count:startDate:)
StationListViewModel
stationList : [StationListItem]
init()
update()
OfferViewModel
offers : [OfferData]
isLoading : Bool
isError : Bool
start : FormStationListItem
end : FormStationListItem
passengerCount : Int
startDate : Date
init(start:end:passengerCount:startDate:)
update()
HomeViewModel
favoriteStations : [FavoriteStationListItem]
recentOffers : [RecentOfferListItem]
alerts : [AlertListItem]
disposables
init()
deleteRecentOffer(id:)
subscribe()
update()
updateFavoriteStations()
updateRecentOffers()
AlertsViewModel
alerts : [Alert]
isError : Bool
isLoading : Bool
disposables
init()
subscribe()
«protocol»
ApiProtocol
shared : any ApiProtocol
cities : Cities
customers : CustomersAndDiscounts?
stationList : StationList
stationLocationList : [StationLocation]
services : ServicesDto?
getOffer(startCode:endCode:passengerCount:startDate:completion:)
getTrainLocations(completion:)
getStationInfo(stationNumberCode:completion:)
getTrainInfo(trainId:completion:)
notifier : PassthroughSubject<(), Never>
«protocol»
StoreProtocol
shared : any StoreProtocol
favoriteStations : [FavoriteStation]
recentOffers : [RecentOffer]
controller : PersistenceController
publisher : PassthroughSubject<StoreFields, Never>
updateFavoriteStationList()
saveFavoriteStation(code:searchCount:isFavorite:)
updateRecentOfferList()
deleteRecentOffer(id:)
saveRecentOffer(startCode:endCode:)
favoriteStationSearchCount(code:)
isFavoriteStation(code:)
«protocol»
RssProtocol
shared : RssProtocol
rssItemList : [RssItem]
publisher : PassthroughSubject<RssFields, Never>
confirms to
confirms to
confirms to
ApiRepository
shared
cities : Cities
customers : CustomersAndDiscounts?
stationList : StationList
services : ServicesDto?
stationLocationList : [StationLocation]
isLoading : Bool
isError : Bool
cancellableSet : Set<AnyCancellable>
notifier
init()
update()
updateCities()
updateCustomersAndDiscounts()
updateStationList()
updateServices()
getOffer(startCode:endCode:passengerCount:startDate:completion:)
getStationLocation()
getTrainLocations(completion:)
getStationInfo(stationNumberCode:completion:)
getTrainInfo(trainId:completion:)
StoreRepository
shared
favoriteStations : [FavoriteStation]
recentOffers : [RecentOffer]
controller : PersistenceController
publisher
init()
notify()
updateFavoriteStationList()
saveFavoriteStation(code:searchCount:isFavorite:)
updateRecentOfferList()
deleteRecentOffer(id:)
saveRecentOffer(startCode:endCode:)
favoriteStationSearchCount(code:)
isFavoriteStation(code:)
RssRepository
shared
rssItemList : [RssItem]
isError : Bool
isLoading : Bool
lastTag : String?
lastContent : String?
parseList : [RssItem]
publisher
init()
update()
startParsing(inputData:)
handleNewTag()
parser(_:foundCharacters:)
parser(_:didStartElement:namespaceURI:qualifiedName:attributes:)

Classes

ApiRepository

RssRepository

Class parsing RSS feed

StoreRepository

Store repository to load inital data and to proxy requests to CoreData

RssItem

Data type representing an RSS feed item

UniqueObjectWithId

Data type for easy conformance with Identifiable, Equatable, and Hashable protocols

TextFieldDebouncer

Class handling debouncing (delaying) of a TextField's input

Theme

CustomMapAnnotation

Custom annotation view using MapIcon SwiftUI component

MapKitMap.Coordinator

Internal class for MapKit compatibility

AlertsViewModel

Alert View Model for Alert List

DirectionsFormViewModel

Form view model

HomeViewModel

Home view model with favorite stations, recent offers and alerts

LoadingViewModel

Loading view model to handle initial API calls, wait for them and indicate error or loading

LocationItem

Location item for MapKit

MapViewModel

Map view model for the main map

OfferViewModel

Offer view model for result screen

StationDetailsViewModel

Station details view model

StationListViewModel

Station list view model with every station the API returns

TrainViewModel

Train details view model

Structures

MavPlusApp

PersistenceController

StationLocation

Station location with code and coordinates

ApiRepositoryFields

RssFields

Data type for the result of RSS parsing

StoreFields

Store fields main data type

City

Data type representing a city

CustomersAndDiscounts

Data type representing discounts and customer types

CustomerType

Data type representing the different types of customers

DiscountProp

ForeignDescriptions

Data type representing the description of CustomerType objects in foreign languages

Foreign

Data type representing the name of CustomerType objects in foreign languages

Discount

Data type representing the types of discounts

DiscountValue

Data type representing the value of a discount type

CustomersQueryDto

Offer

Data type representing a direction offer

OfferRoute

Data type representing the parts of an Offer

Arrival

The time data of the departure/arrival of a train at a station

Track

Data type representing the arrival/departing track of a train at a station

Details

Data type representing details of a part of a route (OfferRoute)

DetailsRoute

Details of a part of a route in Details

DestionationStationClass

Data for a station along a part of a route

MasodlagosEszkozSzolgaltatasok

Services

The list of services on the train, and along a part of a route

RouteService

TrainDetails

Details about a train

TrainKind

Data type with the kind of a train

EndStationClass

Data about a starting/ending station of a train

TravelClass

Data about a travel class on a train

Price

Price of a TravelClass on a train

PriceCurrency

Currency associated with a Price

Ticket

Data type of a ticket

Params

ClientDiscount

A discount a client has

DiscountedGrossPrice

Gross price after a discount

DiscountedGrossPriceCurrency

Currency of a discounted price

PlaceTicket

Seat reservation data

TicketTrain

Train associated with a ticket

OfferRequestQueryDto

Data Transfer Object (DTO) for an Offer request

Passenger

Data type representing passengers

ServicesDto

Data Transfer Object (DTO) for services

SearchService

Data type for search parameters

ForeignNames

Name of a service/parameter in foreign languages

Service

Data type for services

ServiceQueryDto

StationInfo

Data on a station

StationSchedulerDetails

Station scheduler

Scheduler

Schedule data

FullNameAndPiktogram

Full name and icon of a train

HavarianInfok

Additional information in case of a service disturbance

Kind

Sign

An icon

StationService

ViszonylatObject

Data on a route

ViszonylatiJel

Icon for a route

StationInfoQueryDto

Data Transfer Object (DTO) for station info queries

Station

Data on a station

TrainInfo

Data on train schedules

TrainSchedulerDetail

Data for a scheduled arrivals/departures of a train

TrainScheduler

Scheduled arrival/departure of a train at a station

Train

Data on a train

TrainInfoQueryDto

Data Transfer Object (DTO) for train info queries

TrainLocationList

List of train locations

TrainLocation

Data on a train location

Viszonylat

Data on a route for trains

LocationQueryDto

Data Transfer Object (DTO) for a location query

WarninMessagesDto

Data Transfer Object (DTO) for warning messages

ErrorView

Error screen.

ErrorView_Previews

SwiftUI Preview

IconField

A UI component with text and an image

IconField_Previews

SwiftUI Preview

MapIcon

An icon on the map with an image.

MapIcon_Previews

SwiftUI view generation.

MapKitMap

SwiftUI embeddable UIKit based Map view

MapKitMap_Previews

SwiftUI Preview

SpinnerView

Spinning loading icon UI component

Spinner

Model for custom loading UI component

SpinnerCircle

Spinner_Previews

SwiftUI Preview

WebView

SwiftUI embeddable UIKit based web view component

WebView_Previews

SwiftUI Preview

DirectionsOfferListItem

View for an Offer item in the list of Offers

DirectionsOfferView_Previews

SwiftUI Preview

DirectionsQuerySummary

View for the summary of the query

DirectionsQuerySummary_Previews

SwiftUI Preview

StationPickerField

StationPickerModal

StationPicker_Previews

SwiftUI Preview

TrainListItemData

Data class representing a train in the trainlist.

TrainListItem

View component representing a single train in the trainlist.

TrainListItem_Previews

SwiftUI Preview

RootNavigation

Main TabView component containing every screen of tha app.

RootNavigation_Previews

SwiftUI Preview

DirectionsFinal

View presenting a single directions Offer

DirectionsFinal_Previews

SwiftUI Preview

DirectionsFormScreen

View for direction query input, with TextInputs

Directions_Previews

SwiftUI Preview

DirectionsResultScreen

List view showing the results for a direction query

Home

Home_Previews

HomeScreen

View for the landing screen

Home_Previews

SwiftUI Preview

LoadingScreen

View presenting a loading screen with a spinning loading icon

LoadingScreen_Previews

SwiftUI Preview

MavMap

Custom map for showing stations and trains using MapKit Map, refreshing every 5 seconds

Settings

Placeholder view for settings

Settings_Previews

SwiftUI Preview

StationDetailsMapScreen

Map view showing the location of a single station

StationDetailsScreen

View showing the details of a station (e.g. location, name, train departures, etc.)

DepartureListItem

StationDetailsScreen_Previews

StationListScreen

View listing every station

StationListView_Previews

SwiftUI Preview

TrafficNews

View for RSS feed showing traffic news

TrafficNews_Previews

SwiftUI Preview

TrainDetailsScreen

View presenting the details of a train (e.g. name, stops, how late it is, etc.)

TrainDetailsScreen_Previews

SwiftUI Preview

Alert

Alert list item with url

FormStationListItem

Station picker data model

FavoriteStationListItem

Favorite stations

RecentOfferListItem

Recent offers

AlertListItem

Alerts

OfferData

Offer data

OfferRoutePart

Offer Route Part between two stations in a route, there may be multiple route parts in an offer

StationDetails

Station details

TrainPictogram

Train pictogram with color and name

Departure

Departure with UUID

StationListItem

Station list item data struct

TrainData

Main data type for trains

Stop

Where the train stops and when

Enumerations

InterestingTag

Enum for the tags to parse

FullNameAndPiktogram.CodingKeys

Protocols

RssProtocol

Protocol for accessing RssRepository

RequestStatus

Protocol for a request's status

Updateable

Protocol for updatable classes

Typealiases

Cities

StationList

Functions

citiesRequest(completion:)

Gets the cities with postal codes

customersRequest(completion:)

Gets customer types and dicount types

stationListRequest(completion:)

Gets the station list

servicesRequest(completion:)

Gets available services to search for in offers

trainInfoRequest(trainId:completion:)

Gets data of a single train

stationInfoRequest(stationNumberCode:completion:)

Gets data of a single station

offerRequest(startCode:endCode:passengerCount:startDate:completion:)

Get offer by start, end, passenger count and start date

trainLocationRequest(completion:)

Get train location list (for every train)

getOffer(startCode:endCode:passengerCount:startDate:completion:)

Get offer by start, end, passenger count and start date

getTrainLocations(completion:)

Get train location list (for every train)

getStationInfo(stationNumberCode:completion:)

Get info for a single station

getTrainInfo(trainId:completion:)

Get info for a single train

updateFavoriteStationList()

Retrieves every saved favorite station

saveFavoriteStation(code:searchCount:isFavorite:)

Saves a station as favorite

updateRecentOfferList()

Retrieves every saved recent offer

deleteRecentOffer(id:)

Deletes a recent offer by ID

saveRecentOffer(startCode:endCode:)

Saves recent offer

favoriteStationSearchCount(code:)

Retrieves station search count

isFavoriteStation(code:)

Checks whether the station with the given code is saved as favorite

DateFromIso(_:)

Conveniance function converting an ISO8601 date as a String to Date object

Variables

ContentType

Accept

Language

UserSessionId

shared

Singleton instance

cities

Cities with postal codes

customers

Customer types and discounts

stationList

Station list

stationLocationList

Station locations

services

Search services

notifier

Notifies whether data is available

shared

Singleton instance of the class

favoriteStations

Favorite stations without any additional data

recentOffers

Recent offers without any additional data

controller

Persistence controller for CoreData

publisher

Async publisher, which sends when the data is retrieved from the storage

MockStationList

MockTrainPictogram

MockOfferRoutePart

MockOfferData

MockTrainData

MockError

MockQueryStartName

MockQueryEndName

MockQueryPassengerCount

MockQueryStartDate

CitiesRequestPath

URL for requests of cities

CustomersRequestPath

URL for requests of customers

OfferRequestPath

URL for requests of offers

ServicesRequestPath

URL for requests of services

TimetableRequestPath

URL for requests of schedules

StationsRequestPath

URL for requests of stations

WarningMessagesRequestPath

URL for requests of warning messages

TrainLocationRequestPath

URL for requests of train locations

RssFeedRequestPath

URL for requests of RSS feed

clusterID

Extensions

Color

Generated on December 7, 2022 using swift-doc 1.0.0-rc.1.