Apod

+ Apod():

- lowDefUrl: URL
- mediaType: MediaType
- dateTaken: LocalDate
- isFavorite: boolean
- copyright: String
- id: long
- description: String
- location: String
- date: LocalDate
- title: String
- highDefUrl: URL

description: String
highDefUrl: URL
dateTaken: LocalDate
location: String
title: String
isFavorite: boolean
date: LocalDate
copyright: String
lowDefUrl: URL
mediaType: MediaType
id: long

ApodDatabase

+ ApodDatabase():

- NAME: String

apodDao: ApodDao
NAME: String

Converters

+ Converters():

+ integerToMediaType(Integer): MediaType
+ urlToString(URL): String
+ mediaTypeToInteger(MediaType): Integer
+ localDateToLong(LocalDate): Long
+ longToLocalDate(Long): LocalDate
+ stringToUrl(String): URL

DayBinder

+ DayBinder(Context):

- listener: OnApodClickListener
- apodMap: Map<LocalDate, Apod>

+ create(View): ViewContainer
+ bind(ViewContainer, CalendarDay): void

listener: OnApodClickListener
apodMap: Map<LocalDate, Apod>

ImageFragment

+ ImageFragment():

+ onDestroyView(): void
+ onViewCreated(View, Bundle?): void
+ onCreate(Bundle?): void
+ onCreateView(LayoutInflater, ViewGroup?, Bundle?): View?
- handleApod(Apod): void

ImageCallback

+ ImageCallback():

+ onSuccess(): void
+ onError(Exception): void

<<interface>>
OnApodClickListener

+ onApodClick(Apod): void

DayHolder

+ DayHolder(View):

- translateClick(View): void
+ bind(CalendarDay): void

<<enumeration>>
MediaType

- MediaType(boolean):

- downloadable: boolean

+ values(): MediaType[]
+ valueOf(String): MediaType

ApodViewModel

~ ApodViewModel(ApodRepository):

- apodId: MutableLiveData<Long>
- apods: LiveData<List<Apod>>
- yearMonth: MutableLiveData<YearMonth>

- transformYearMonthToQuery(YearMonth): LiveData<List<Apod>>
- postThrowable(Throwable): void

apods: LiveData<List<Apod>>
apodMap: LiveData<Map<LocalDate, Apod>>
apod: LiveData<Apod>
yearMonth: LiveData<YearMonth>
apodId: long

HeaderBinder

~ HeaderBinder(Context):

+ bind(ViewContainer, CalendarMonth): void
+ create(View): ViewContainer

HeaderHolder

+ HeaderHolder(View):

+ bind(CalendarMonth): void

CalendarFragment

+ CalendarFragment():

+ onViewCreated(View, Bundle?): void
- handleApods(Map<LocalDate, Apod>): void
- showApod(Apod): void
+ onCreateView(LayoutInflater, ViewGroup?, Bundle?): View?
- handleYearMonth(YearMonth): void
+ onDestroyView(): void
- handleMonthScroll(CalendarMonth): Unit
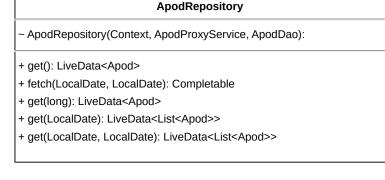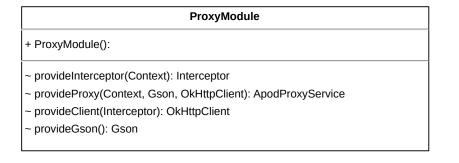
<<interface>>
ApodDao

+ selectRange(LocalDate, LocalDate): LiveData<List<Apod>>
+ insert(Apod): Completable
+ insert(List<Apod>): Completable
+ select(LocalDate): LiveData<Apod>
+ selectOpenRange(LocalDate): LiveData<List<Apod>>
+ deleteAll(): Completable
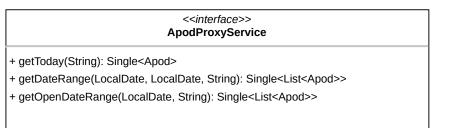+ select(long): LiveData<Apod>

andUpdateFavorites: LiveData<List<Apod>>

ApodRepository

~ ApodRepository(Context, ApodProxyService, ApodDao):

+ get(): LiveData<Apod>
+ fetch(LocalDate, LocalDate): Completable
+ get(long): LiveData<Apod>
+ get(LocalDate): LiveData<List<Apod>>
+ get(LocalDate, LocalDate): LiveData<List<Apod>>

ProxyModule

+ ProxyModule():

~ provideInterceptor(Context): Interceptor
~ provideProxy(Context, Gson, OkHttpClient): ApodProxyService
~ provideClient(Interceptor): OkHttpClient
~ provideGson(): Gson

MainActivity

+ MainActivity():

+ onSupportNavigateUp(): boolean
# onCreate(Bundle): void
- setupNavigation(): void
- setupUI(): void

DatabaseModule

+ DatabaseModule():

~ provideApodDao(ApodDatabase): ApodDao
~ provideDatabase(Context): ApodDatabase

<<interface>>
ApodProxyService

+ getToday(String): Single<Apod>
+ getDateRange(LocalDate, LocalDate, String): Single<List<Apod>>
+ getOpenDateRange(LocalDate, String): Single<List<Apod>>

PicassoModule

+ PicassoModule():

~ providePicasso(Context): Picasso

NasaApodApplication

+ NasaApodApplication():

+ onCreate(): void