National Software Development Framework

Anar Framework Architecture Technical Documentation

Big Data Directorate Team

Introduction

Nowadays, for governmental and any other large systems reliability, sustainability, and interoperability are vital, without these features the systems would not survive. Best practices and design principles for architectural design and development have to be followed and considered. The best and proven way is to build a collection of integrated libraries and modules under a framework to produce a well-designed architecture and implementation.

In recognition that there is a clear need for a standardized development framework and tools necessary to implement governmental application services, reduce vendor dependency, increase efficiency, enhance the interoperability of systems, resolve dependency on outsider vendors and enhance the software quality, we have invested in building our own national software development framework (Anar Framework). The framework provides the foundation and tools for developing software applications that allow software developers to focus only on the business logic, rather than the technical infrastructure of a system. The framework consists of pre-configured templates and predefined guidelines for contribution developer in a project. T the templates take care of non-functional requirements, best practices of creating a new service, and other common functionalities for developers. The developers just need to pay attention on business logics and domain driven implementations. In the following there are only few advantages that the Anar Framework provides.

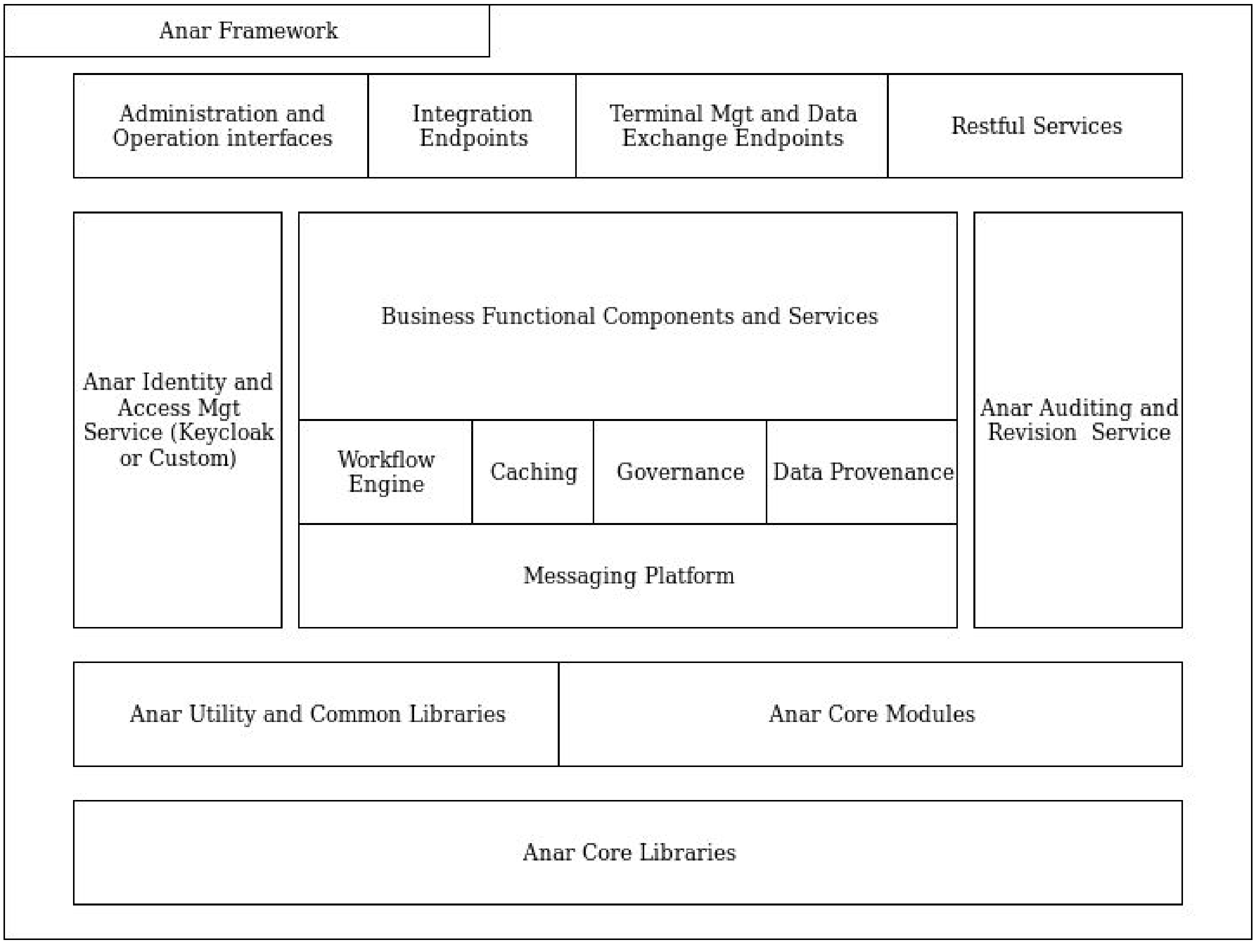
- Provides common essential functions and utility classes for building well-designed systems.
- Provides a per-configured, integrated, and comprehensive development environment.
- Provides general technical infrastructure of software applications, allowing developers to only focus on writing code for the business logic.
- Simplifies software development process by providing standardized tools for developing software and built-in decisions about fundamental architecture and design issues
- Improves productivity by increasing the reusability of common components of systems developed through the standard framework, thereby avoiding redundant development of same components
- Raises quality of software applications by providing technical code for generic services needed by most applications, where developers are prone to making errors
- Standardizes the software development process for consistency and predictability
- Enhances interoperability of systems and information-sharing between related department through standard inter-system integration interfaces.

- Provides standards for modular programming and thereby facilitates swift replacement of modules for easy maintenance of software applications and adoption of fast changing technologies
- Provides a reliable technology infrastructure based on carefully selected Open Source and major software vendors' frameworks

Architecture

Anar Framework mainly consists of following parts:

- Anar core, utility and common libraries
- Anar core modules and services
- Anar Identity and Access Management
- Anar Auditing, Revision, and Data provenance
- Workflow Engine
- Messaging Platform Integration
- Business functional components and services
- Administration and Operation Interfaces
- Integration Endpoints


Anar Framework
Administration and Operation interfaces
Integration Endpoints
Terminal Mgt and Data Exchange Endpoints
Restful Services
Anar Identity and Access Mgt Service (Keycloak or Custom)
Business Functional Components and Services
Workflow Engine
Caching
Governance
Data Provenance
Messaging Platform
Anar Auditing and Revision Service
Anar Utility and Common Libraries
Anar Core Modules
Anar Core Libraries

Anar core, utility and common libraries

The framework consists of 50+ libraries that each one of them are hosted on a separate repository with detailed documentation in Github.com (https://github.com/Anar-Framework).

anar-lib-core

This library is one of the core libraries and contains the features that mostly used in all other libraries and modules. These are the main features that the library provides:

- Services Naming

- Versioning
- Common Annotations
- Exception Handling
- Error Handling
- Common Validation

anar-lib-api

This library is also one of the core libraries and contains the features that mostly used in all modules and services. These are the main features that the library provides:

- ThrowException (s) Annotations
- Enable Api Factory Annotation and Configuration
- User Context
- Cookie Management
- Common Exceptions
- Response Handler - controller Advice

anar-lib-json

This is a utility library and these are the main features that the library provides:

- converts the JSON String input and maps it to the java object
- returns the value associated with the label of the input JSON
- converts a JSON String containing multiple JSON and stores them in a java list
- converts a JSON String containing multiple JSON and stores them in a java Map

anar-lib-xml

This is a utility library and these are the main features that the library provides:

- Create new XML documents from external sources or new document from scrash
- Manage namespaces
- Manipulating nodes (add, remove, rename)
- Manipulating data (add, remove text or CDATA)

- Navigate into the document with shortcuts and XPath (note: XPath supports namespaces)
- Transform an XMlDoc instance to a String or a Document
- Validate your document against schemas
- Executing callbacks on a hierarchy
- Remove all namespaces (namespace ignoring)

anar-lib-excel

This is a utility library and these are the main features that the library provides:

- Cell Template
- Row Template
- Style Template
- Text Cell
- Numeric Cell
- Empty Cell
- Formula Cell
- Date Cell
- Calendar Cell
- Flexible for adding more types of cells
- Vertical Alignment
- Horizontal Alignment
- Background Color
- Border Bottom
- Border Left
- Border Right
- Border Top
- Date Format
- Fill Pattern
- Font Style
- Foreground Color
- Hidden
- Indention
- Lock
- Rotation
- Wrap Text
- Row Height
- Row Height In Points

- Row Zero Height
- Sheet Auto Size Column
- Sheet Column Width
- Sheet Default Column Width
- Sheet Row Height
- Sheet Fit To Page
- Sheet Merge Region

anar-lib-html

This is a utility library and these are the main features that the library provides:

- Removes all HTML element tags from a string, leaving just the character data
- Escape characters that are special in HTML, so that the resulting string can be included in HTML (or XML)
- Converts all inline HTML character entities
- Converts appropriate Unicode characters to their HTML character entity counterparts
- Convenience method to convert embedded HTML to text

anar-lib-math

This is a utility library and these are the main features that the library provides:

- Raise an int to an int power.
- Generate random number
- Compute the square root of a number.
- Compute the maximum of two values
- Rounds the given value to the specified number of decimal places.
- Absolute value.
- Evaluate Factorial
- Computes the greatest common divisor of the absolute value of two numbers, using a modified version of the "binary gcd" method.
- Natural logarithm.
- Compute the base 10 logarithm.
- Creates a copy of source array
- Returns the maximum of the entries in the input array, or Double.NaN if the array is empty.
- Returns the minimum of the entries in the input array, or Double.NaN if the array is empty.

- Returns the sum of the values in the input array, or Double.NaN if the array is empty.
- getValuesProduct: Returns the product of the entries in the input array, or Double.NaN if the array is empty.
- Returns the arithmetic mean of the entries in the input array, or Double.NaN if the array is empty.
- Primality test: tells if the argument is a (provable) prime or not.
- Return the smallest prime greater than or equal to n.
- Prime factors decomposition

anar-lib-date

Utilities for Date Time operations. Provide Date and Time utility for usage across the application to manipulate dates . This is a utility library and these are the main features that the library provides:

- Adds a number of days to a date returning a new Date object.
- Adds a number of hours to a date returning a new Date object.
- Adds a number of minutes to a date returning a new Date object.
- Adds a number of seconds to a date returning a new Date object.
- Formats a date/time into a specific pattern.
- Formats a date/time into a specific pattern in a time zone.
- Formats a date/time into a specific pattern in a time zone and locale.
- Formats a calendar into a specific pattern.
- Formats a calendar into a specific pattern in a time zone.
- Formats a calendar into a specific pattern in a time zone and locale.
- Formats a calendar into a specific pattern in a time zone and locale.

anar-lib-calendar

This is a utility library and these are the main features that the library provides:

- Get Ceiling : Gets a date ceiling, leaving the field specified as the most significant field.
- Get Fragment In Day: Returns the number of days within the fragment. All datefields greater than the fragment will be ignored.
- Get Fragment In Hour: Returns the number of hours within the fragment. All datefields greater than the fragment will be ignored.

- Get Fragment In Millisecond: Returns the number of milliseconds within the fragment. All datefields greater than the fragment will be ignored.
- Get Fragment In Minutes: Returns the number of minutes within the fragment. All datefields greater than the fragment will be ignored.
- Get Fragment In Seconds: Returns the number of seconds within the fragment. All datefields greater than the fragment will be ignored.
- In Same Day: Checks if two calendar objects are on the same day ignoring time.
- Is Same Instant: Checks if two calendar objects represent the same instant in time. This method compares the long millisecond time of the two objects.
- Is Same Local Time :Checks if two calendar objects represent the same local time. This method compares the values of the fields of the two objects. In addition, both calendars must be the same of the same type.
- get Round : Rounds a date, leaving the field specified as the most significant field.
- to Calender: Converts a Date of a given TimeZone into a Calendar.
- Truncate : Truncates a date, leaving the field specified as the most significant field.
- Truncate Equals: Determines if two calendars are equal up to no more than the specified most significant field.

anar-lib-mail

This is a utility library and these are the main features that the library provides:

- Flexible Config Builder with properties file
- Mail Builder
- Send Mail Simple and File-attached emails
- Multiple receivers support
- CC and BCC support
- Gmail Mail Builder

anar-lib-str

This is a utility library and these are the main features that the library provides:

- capitalizeFirstLatter: This method capitalizes the first character of the passed parameter
- compare: This method compares two input string
- compareIgnoreCase: This method compares two Strings lexicographically, ignoring case differences
- contains : This method checks if CharSequence contains a search CharSequence, handling null

- containsIgnoreCase :This method checks if CharSequence contains a search CharSequence irrespective of case, handling null.
- containsWithWhiteSpace : This method checks whether the given CharSequence contains any whitespace characters
- isEmpty : This method checks if a CharSequence is empty ("") or null -isNotEmpty : This method checks if a CharSequence is not empty ("") and not null
- isBlank: This method checks if a CharSequence is empty (""), null or whitespace only
- isNotBlank: This method checks if a CharSequence is not empty (""), not null and not whitespace only
- trim : This method removes control characters (char is less than= 32) from both ends of this String, handling null by returning null
- truncate : Truncates a String
- strip: Strips whitespace from the start and end of a String
- equals : Compares two CharSequences
- equalsIgnoreCase: Compares two CharSequences ignoring case
- indexOf : Returns the index within seq of the first occurrence of the specified character
- lastIndexOf :Returns the index within seq of the last occurrence of the specified character
- substring: Gets a substring from the specified String avoiding exceptions.
- removeLeftChar: Gets the leftmost len characters of a String
- removeRightChar: Gets the rightmost len characters of a String
- removeMidChar: Gets len characters from the middle of a String
- substringBefore: Gets the substring before the first occurrence of a separator
- substringAfter: Gets the substring after the first occurrence of a separator
- split: Splits the provided text into an array, using whitespace as the separator
- splitByCharType: Splits a String by Character type
- join: Joins the elements of the provided array into a single String containing the provided list of elements
- deletewhiteSpace: Deletes all whitespaces from a String
- remove: Removes all occurrences of a substring from within the source string
- removeIgnoreCase: Case insensitive removal of all occurrences of a substring from within the source string
- removeAll: Removes each substring of the text String that matches the given regular expression
- replaceOnce: Replaces a String with another String inside a larger String, once
- replaceIgnoreCase: Case insensitively replaces a String with another String inside a larger String, once
- replacePattern: Replaces each substring of the source String that matches the given regular expression with the given replacement

- removePattern: Removes each substring of the source String that matches the given regular expression
- replaceAll: Replaces each substring of the text String that matches the given regular expression with the given replacement
- replaceChar: Replaces all occurrences of a character in a String with another
- overley: Overlays part of a String with another String
- chomp: Removes one newline from end of a String if it's there, otherwise leave it alone
- repeat: Repeat a String repeat times to form a new String
- length: String with repeated character
- uncapitilize: Uncapitalizes a String, changing the first character to lower case
- swapCase: Swaps the case of a String changing upper and title case to lower case, and lower case to upper case
- countMatches: Counts how many times the substring appears in the larger string
- isAlpha: Checks if the CharSequence contains only Unicode letters
- isAlphaSpace: Checks if the CharSequence contains only Unicode letters and space (' ')
- isAlphanumeric: Checks if the CharSequence contains only Unicode letters or digits
- isAlphanumericSpace: Checks if the CharSequence contains only Unicode letters, digits or space (' ')
- isNumeric :Checks if the CharSequence contains only Unicode digits. A decimal point is not a Unicode digit and returns false
- getDigits: Checks if a String sourceString contains Unicode digits, if yes then concatenate all the digits in sourceString and return it as a String
- isWhiteSpace: Checks if the CharSequence contains only whitespace
- rotate: Rotate (circular shift) a String of shift characters
- reverse: Reverses a String
- reverseDelimited :Reverses a String that is delimited by a specific character
- abbreviate: Abbreviates a String using ellipses
- abbreviateMiddle: Abbreviates a String to the length passed, replacing the middle characters with the supplied replacement String
- difference :Compares two Strings, and returns the portion where they differ
- startsWith: Check if a CharSequence starts with a specified prefix
- endswith: Check if a CharSequence ends with a specified suffix
- normalizeSpace: Returns the argument string with whitespace normalized
- appendIfMissing: Appends the suffix to the end of the string if the string does not already end with any of the suffixes
- prependIfMissing: Prepends the prefix to the start of the string if the string does not already start with any of the prefixes
- wrap: Wraps a string with a char

- unwrap: Unwraps a given string from anther string

anar-lib-uuid

This is a utility library and these are the main features that the library provides:

- takes UUID namespace and a name and generate Type 5 UUID.
- takes UUID namespace and a name as a byte array and generate Type 5 UUID.

anar-lib-velocitytemplate

Velocity is a template engine that can be used for many purposes. Some common types of applications which use Velocity are:

- Web applications. Web designers create HTML pages with placeholders for dynamic information. The page is processed with VelocityViewServlet or any of a number of frameworks which support Velocity. This approach to web application development is called Model-View-Controller or MVC and is intended to be a direct replacement for applications developed with Java Server Pages (JSPs) or PHP.
- Source code generation. Velocity can be used to generate Java source code, SQL, or PostScript based on templates. The PoweredByVelocity page lists a number of open source and commercial development software packages which use Velocity in this manner.
- Automatic emails. Many applications generate automatic emails for account signup, password reminders or automatically sent reports. Using Velocity, the email template can be stored in a text file rather than directly embedded in your Java code.
- XML transformation. Velocity provides an ant task called Anakia which reads an XML file and makes it available to a Velocity template. A common application is to convert documentation stored in a generic "xdoc" format into a styled HTML document.

anar-lib-zip

Utilities for Zip and UnZip operations. Provide Zip utility for usage across the application to Zip and unZip Files and Directory Size of the files and Folders according to business needs This ZipUtil will not applicable for RAR or 7Z etc. This is a utility library and these are the main features that the library provides:

- zipping a Byte Array
- unzipping a zipped Byte Array
- read a file
- zip all files of the given Directory
- UnZipFile used for created output folder
- Extracts a zip file specified by the zipFilePath to a directory specified by destDirectory (will be created if does not exist)
- Extracts a zip entry (file entry)

anar-lib-hsm

This is a utility library for integration with Hardware Security Module (HSM) and these are the main features that the library provides:

- Calculate CVV
- Validate CVV
- calculate Offset Using Pin
- calculate Offset Using Pin LMK
- calculate Offset Using Pin Block
- change Pin Offset
- validate Terminal Pin
- validate Interchange Pin
- validate DUKPTPin
- calculate KCV
- generate Key
- generate Export Key
- derive IPEK
- derive Export IPEK
- form Key From Components
- import Key
- export Key
- calculate MAC
- validate MAC
- calculate PVV Using Pin
- calculate PVV Using Pinblock
- validate DUKPT Pin
- validate Interchange Pin

- validate Terminal Pin
- change PVV
- encrypt Pin Under LMK
- encrypt Pinblock Under LMK
- decrypt Pin Under LMK
- Translate fromZPKToZPK
- Translate fromTPKToZPK
- Translate fromTPKToBDK
- Translate fromBDKToBDK
- Translate fromBDKToZPK
- Translate fromZPKToLMK
- Translate fromTPKToLMK
- Translate fromLMKToZPK

anar-lib-pdf

This is a utility library and these are the main features that the library provides:

- Convert HTML obtained from an InputStream to a PDF written to an OutputStream.
- Convert process Template as String to outpustStream
- Take input as template file and convert template to PDF and save to the given path with given fileName.
- Convert InputStream to OutputStream.
- Convert BufferedImage list to Byte Array.
- Merge PDF files.
- Convert data obtained from an InputStream to a password protected PDF written to an OutputStream. If password is null or empty, PDF will not be encrypted.

anar-lib-reportutil

This is a utility library and these are the main features that the library provides:

- Report Entity Annotations
- Annotation Parser Configs

anar-lib-csv

This library preserves many of the nice configurable options, changes a few to be more consistent or sensible and adds a number of new ones. This is a utility library and these are the main features that the library provides:

- CSV Parser
- CSV Reader
- CSV Writer
- CSV to Bean Convertor
- Bean to CSV Convertor
- Configurable Separator
- Database Dumping
- Rich configurable properties

anar-lib-checksum

This utility library provides functions to generate Verhoeff's checksum and validate Verhoeff's checksum compliance and these are the main features that the library provides:

- Generates the Verhoeff checksum digit for the provided numeric string.
- Validates that an entered number is Verhoeff checksum compliant. Make sure the check digit is the last one.
- Converts a string to a reversed integer array.
- Reverses an int array.

anar-lib-cors

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served. A web page may freely embed cross-origin images, stylesheets, scripts, iframes, and videos. This library to manage CORS and these are the main features that the library provides:

- Manage allowed origins
- Manage allowed ports
- Manage allowed header

anar-lib-sms

This library is to integrate with the Kannel Open Source WAP and SMS gateway works as both an SMS gateway, for implementing keyword based services via GSM text messages, and a WAP gateway, via UDP. The SMS part is fairly mature, the WAP part is early in its development. In this release, the GET request for WML pages and WMLScript files via HTTP works, including compilation for WML and WMLScript to binary forms. Only the data call bearer (UDP) is supported, not SMS.. This library to manage SMS and these are the main features that the library provides:

- Flexible configuration with properties
- GSM networks support
- Client logs into the SMS center
- SMS management
- SMS log store

anar-lib-file

his is a utility library and these are the main features that the library provides:

- byteCoundToDisplaySize :Returns a human-readable version of the file size, where the input represents a specific number of bytes.
- checksum: Computes the checksum of a file using the specified checksum object
- checksumCRC32: Computes the checksum of a file using the CRC32 checksum routine
- cleanDirectory: Cleans a directory without deleting it
- contentEquals: Compares the contents of two files to determine if they are equal or not
- contentEqualsIgnoreEOL: Compares the contents of two files to determine if they are equal or not. This method checks to see if the two files point to the same file, before resorting to line-by-line comparison of the contents.
- convertFileCollectionToFileArray: Converts a Collection containing java.io.File instanced into array representation
- copyDirectory: Copies a whole directory to a new location preserving the file dates
- copyFile: Copies a file to a new location preserving the file date
- copyInputStreamTofile: Copies bytes from an InputStream source to a file destination.
- copyTofile: Copies bytes from an InputStream source to a file destination The source stream is left open
- deleteDirectory: Deletes a directory recursively
- deleteQuietly: Deletes a file, never throwing an exception

- directoryContains :Determines whether the parent directory contains the child element (a file or directory)
- forceDelete: Deletes a file.
- forceDeleteOnExit: Schedules a file to be deleted when JVM exits
- getFile :Construct a file from the set of name elements.
- isFileNewer: Tests if the specified File is newer than the specified Date
- isFileOlder :Tests if the specified File is older than the specified Date
- isSymlink: Determines whether the specified file is a Symbolic Link rather than an actual file
- iterateFiles: Allows iteration over the files in given directory (and optionally its sub directories). All files found are filtered by an IOFileFilter
- lineIterator: Returns an Iterator for the lines in a File using the default encoding for the VM
- listFiles: Finds files within a given directory (and optionally its subdirectories). All files found are filtered by an IOFileFilter.
- listFilesandDirs: Finds files within a given directory (and optionally its subdirectories) All files found are filtered by an IOFileFilter. The resulting collection includes the starting directory and any subdirectories that match the directory filter
- moveDirectory :Moves a directory
- moveDirectoryToDirectory :Moves a directory to another directory
- moveFile :Moves a file
- moveFileToDirectory: Moves a file to a directory
- moveToDirectory :Moves a file or directory to the destination directory
- openInputStream: Opens a FileInputStream for the specified file,
- openOutputStream: Opens a FileOutputStream for the specified file, checking and creating the parent directory if it does not exist.
- readFileToByteArray :Reads the contents of a file into a byte array
- readFileToString: Reads the contents of a file into a String. The file is always closed.
- readLines :Reads the contents of a file line by line to a List of Strings The file is always closed
- sizeOf: Returns the size of the specified file or directory The return value may be negative if overflow occurs
- sizeOfDirectory: Counts the size of a directory recursively (sum of the length of all files)
- toFile: Convert from a URL to a File
- toURL: Converts each of an array of File to a URL.
- waiteFor: Waits for NFS to propagate a file creation, imposing a timeout.
- write: Writes a CharSequence to a file creating the file if it does not exist.
- writeByteArryToFile: Writes a byte array to a file creating the file if it does not exist.

- writeLines :Writes the toString() value of each item in a collection to the specified File line by line. The default VM encoding and the default line ending will be used.
- writeStringToFile: Writes a String to a file creating the file if it does not exist.

anar-lib-hash

This library provides HashCodeBuilder which is referred to build a good hashCode method for any class and these are the main features that the library provides:

- append: Append a hashCode for a boolean. This adds 1 when true, and 0 when false to the hashCode. This is in contrast to the standard java.lang.Boolean.hashCode handling, which computes a hashCode value of 1231 for java.lang.Boolean instances that represent true or 1237 for java.lang.Boolean instances that represent false. This is in accordance with the Effective Java design.
- appendArray: Append a hashCode for an array.
- toHash: Return the computed hashCode.

anar-lib-hmac

The HMAC Utility is implemented using desired methods of MessageDigest class of java security package and these are the main features that the library provides:

- Performs a digest using the specified array of bytes.
- Updates the digest using the specified byte
- Return the digest as a plain text with Salt
- Return the digest as a plain text
- Creates a message digest with the specified algorithm name.
- Generate Random Salt (with default 16 bytes of length).
- Generate Random Salt (with given length)
- Encodes to BASE64 String
- Decodes from BASE64

anar-lib-jobs

This is a utility library and these are the main features that the library provides:

- Annotation that marks a method to be scheduled. Exactly one of the cron, fixedDelay fixedRate attributes must be specified
- can run embedded within another free standing application
- can be instantiated within an application server (or servlet container), and participate in XA transactions
- can run as a stand-alone program (within its own Java Virtual Machine), to be used via RMI
- can be instantiated as a cluster of stand-alone programs (with load-balance and fail-over capabilities) for the execution of jobs
- Jobs can be any Java class that implements the simple Job interface, leaving infinite possibilities for the work your Jobs can perform.
- Job class instances can be instantiated by Quartz, or by your application's framework.
- When a Trigger occurs, the scheduler notifies zero or more Java objects implementing the JobListener and TriggerListener interfaces (listeners can be simple Java objects, or EJBs, or JMS publishers, etc.). These listeners are also notified after the Job has executed.
- As Jobs are completed, they return a JobCompletionCode which informs the scheduler of success or failure. The JobCompletionCode can also instruct the scheduler of any actions it should take based on the success/fail code - such as immediate re-execution of the Job.
- The design includes a JobStore interface that can be implemented to provide various mechanisms for the storage of jobs.
- With the use of the included JDBCJobStore, all Jobs and Triggers configured as "non-volatile" are stored in a relational database via JDBC.
- With the use of the included RAMJobStore, all Jobs and Triggers are stored in RAM and therefore do not persist between program executions - but this has the advantage of not requiring an external database.

anar-lib-crypto

This is a utility library and these are the main features that the library provides:

- Get splitter index for detaching key splitter from key and data
- Encodes to BASE64 URL Safe
- Encodes to BASE64 String
- Decodes from BASE64

anar-lib-crypto-jce

This is a utility library and these are the main features that the library provides:

- Symmetric Key Generator
- Asymmertric Keys Generator
- Exception Handling
- Utility classes for Crypto management
- JWS Factory
- HMAC Utilities

anar-lib-metrics

This is a Monitoring and Metrics Library with Grafana and Prometheus and these are the main features that the library provides:

- Flexible Config with properties
- Prometheus based matrics data generator
- Production Ready health data
- Exposes audit events information for the current application. Requires an AuditEventRepository bean.
- Displays a complete list of all the Spring beans in your application.
- Exposes available caches.
- Shows the conditions that were evaluated on configuration and auto-configuration classes and the reasons why they did or did not match.
- Displays a collated list of all @ConfigurationProperties.
- Exposes properties from Spring’s ConfigurableEnvironment.
- Shows any Flyway database migrations that have been applied. Requires one or more Flyway beans.
- Shows application health information.
- Displays HTTP trace information (by default, the last 100 HTTP request-response exchanges). Requires an HttpTraceRepository bean.
- Displays arbitrary application info.
- Shows the Spring Integration graph. Requires a dependency on spring-integration-core.
- Shows and modifies the configuration of loggers in the application.
- Shows any Liquibase database migrations that have been applied. Requires one or more Liquibase beans.
- Shows ‘metrics’ information for the current application.
- Displays a collated list of all @RequestMapping paths.
- Displays the scheduled tasks in your application.

- Allows retrieval and deletion of user sessions from a Spring Session-backed session store. Requires a Servlet-based web application using Spring Session.
- Lets the application be gracefully shutdown. Disabled by default.

anar-lib-emptycheck

This is a utility library and these are the main features that the library provides:

- check if given obj is null.
- check if the given str is null or an empty string.
- check given collection is null or is Empty.
- check given map is null or is Empty.

Anar core modules and services

anar-spm

SPM Service of Anar Framework to provide the ability to create custom surveys to collect performance measurement data or any additional questionnaire and these are the main features that the module provides:

- Add and management a new survey.
- Add and management a new entry to a survey.
- Add and management a new survey to collect client related data.

anar-datacollection-kit

Data Collection Kit of Anar Framework and these are the main features that the module provides:

- ODK Integration
- ODKX Integration
- Offline Data Collection
- Online Data Collection
- Form Management
- Instance Management
- Sync Endpoint Management
- Form Workflow management
- Identity and Access Management
- Rich Authentication, Authorization and Accounting

- Multi-tenancy support
- Easy-to-use, comprehensive statistics and data manipulation

anar-cache

Cache module of Anar framework and these are the main features that the module provides using Ehcache:

- Cache and Un-cache annotations
- concurrency features are designed for large, high concurrency systems.
- Sensible defaults require no initial configuration.
- utilize multiple terabytes of data storage.
- Expiration policies
- Eviction policies
- Storage engines (on-heap, off-heap, disk)
- Static/Dynamic/Runtime configuration of caches
- Listener interfaces
- Decorator interfaces
- Cache Loaders, Cache Writers
- Exception Handlers
- full support for high performance distributed caching.

anar-servicetemplate

A template for developing new services and modules for Anar framework or other projects. In this template service, we have implemented following features:

- Integration with Keycloak (Identity and Access management)
- Implementation of required infrastructure configurations, utility classes, constants, caching, installation, deployment, enumerations and other
- Parent for development of a new service
- Sample Business logic implementation
- Developers only need to clone the repository and start adding new features. They don’t need to focus on non-functional requirements.

anar-servicetemplate-customidentity

A template for developing new services and modules for Anar framework or other projects. In this template service, we have implemented following features:

- Integration with our custom identity and access management module instead of Keycloak
- Implementation of required infrastructure configurations, utility classes, constants, caching, installation, deployment, enumerations and other
- Parent for development of a new service
- Sample Business logic implementation
- Developers only need to clone the repository and start adding new features. They don't need to focus on non-functional requirements.

Anar Identity and Access Management

anar-keycloak

Identity and Access Management For Modern Applications and Services and these are the main features that the module provides:

- Single-Sign On
- Identity Brokering and Social Login
- User Federation
- Client Adapters
- Admin Console
- Account Management Console
- Support of Standard Protocols
- Authorization Services

anar-identity-custom

Custom Identity and access management of Anar framework. This module provides the following features:

- Based on Domain Driven Design practices

- Data Provenance Support
- Support of OAuth2 and Open ID Connect
- Privilege Based Authorization
- Support of Role and Group
- Session Management
- Encryption and Decryption Management
- Multi-Tenancy Support
- Digest Authentication Support
- Data Initializers
- Data Revision Support
- Api Resource Security
- Data Auditors

Anar Auditing, Revision, and Data provenance

anar-lib-audit

This is a utility library and these are the main features that the library provides:

- System Action Audit
- User Action Audit
- Flexible and configurable event auditor
- Audit Manager and Audit Request and Response Builder
- Enum Based Auditing
- Annotations

anar-lib-logger

This library is also one of the core libraries and contains the features that mostly used in all modules and services. These are the main features that the library provides:

- Console Log Appender
- File Log Appender
- File Rolling Appender
- Logging Utility Functions
- Session Logging

- Logging Factory
- Different Logging Level
- Logging File Parser

Anar Workflow Engine

anar-lib-workflow

This is the workflow engine library that provides the following features:

- Manage workflow in a RDBMS database
- Flexible and easy JSON based workflow
- Configurable Steps
- Transition Support
- Comment required support
- Authority and access management
- Resolutions Support
- Workflow Parser

Messaging Platform Integration

anar-lib-kafka

This library is to facilitate Apache Kafka integration and these are the main features that the library provides:

- Flexible Config with properties
- Generic Producer Class
- Generic Consumer Contract
- Json, String and Byte Serializer/Deserializer

anar-lib-activemq

This library is to facilitate Apache ActiveMQ integration and these are the main features that the library provides:

- Flexible Config with properties
- Generic Producer Class

- Generic Consumer Contract
- Json, String and Byte Serializer/Deserializer

Business functional components and services

These are the services and modules which have been implemented for specific domains for instance accounting, asset management, crm and other services and modules.

Administration and Operation Interfaces

anar-webapptemplate

A template for developing new client side application for Anar framework or other projects. In this template service, we have implemented following features:

- Integration with Keycloak
- Implementation of required non-functional features
- Proxy
- Sample Business logic implementation
- Developers only need to clone the repository and start adding new features. They don't need to focus on non-functional requirements.

anar-keycloak-theme

This theme is built for the applications which are using Keycloak as identity and access management controller. These are the parts of the theme:

- Login Theme
- Account Theme
- Admin Theme
- Mail Theme
- Welcome Page Theme
- Common Resources

Integration

anar-connect

The module supports a variety of change data capture strategies, both batch and real-time, enabling a entities to select an update strategy that optimizes the overarching data integration processes. This is especially important when data needs to be copied from one or many data-sources to one or many other data-sources or to an analytics data warehouse without disrupting the regular flow of data, which is the case when users are forced to wait for batch runs. CDC streamlines modern analytics by leveraging event-driven data and making data integration more agile to deliver increased operational efficiency and these are the main features that the module provides:

- Create a new connector, returning the current connector info if successful.
- Get information about the connector.
- Get the configuration for the connector.
- Get the status of specified connector by name.
- Update a connector's configuration.
- Restart a connector.
- Pause a connector.
- Resume a connector.
- Get a list of tasks currently running for the connector.
- Get a task’s status.
- Restart an individual task.
- Return a list of connector plugins installed in the Kafka Connect cluster.
- Validate the provided configuration values against the configuration definition. This API performs per config validation, returns suggested values and error messages during validation.

Conclusion

As the sustainability, reliability, interoperability, architectural design and security are becoming essential for governmental systems, it is vital to build a framework (an integrated collection of libraries and modules that collaborate to produce a reusable architecture for building sustainable applications).

To increase efficiency, enhance the interoperability of systems, resolve dependency on outsider vendors and enhance the software quality, we have invested in building our own national software development framework (Anar Framework). The framework provides the foundation and tools for developing software applications that allow software developers to focus only on the business logic, rather than the technical infrastructure of a system. The framework consists of pre-configured templates and predefined guidelines for contribution developer in a project. T the templates take care of non-functional requirements, best practices of creating a new service, and other common functionalities for developers. The developers just need to pay attention on business logics and domain driven implementations.