

API Design

Ridi Ferdiana | ridi@acm.org

Version 1.0.0



The Codes is right
Until...

slow defects
anomaly untraceable



'Good' Code Design is like Insurance



Demo

Code Analytics



API (Application Programming Interface) provides a 'socket' to communicate between component or system

What is API Design?



Abstracts implementation

Eliminating the replication

Providing Interaction

API Characteristic

Understandable

Documented

Consistent

Fit for Purpose

Restrained

Evolvable

Understandable

- Clear entry points
- Discover benefits and use of API
- Minimum External Documentation

Azure SDK for Java API Reference <small>version Stable</small>	
Name	Description
com.azure.ai.formrecognizer Package	Package containing classes for creating a FormRecognizerClient to perform operations on Azure Form Recognizer
com.azure.ai.formrecognizer.models Package	Package containing classes for FormRecognizerClient. Extracts information from forms and images into structured data.
com.azure.ai.formrecognizer.training Package	Package containing form recognizer training clients for Azure Form Recognizer.
com.azure.ai.formrecognizer.training.models Package	Package containing model classes for FormTrainingClient.

Consistency (1/2)

Consistency in Class Name, Method, Arguments, Constants, and Process

A minimal Set of Return Types

Avoid Surprise (Inconsistency method in similar classes)

Returning non-null values instead (Defensive Programming)

- String – ""

- List - .emptyList()

- Stream - .empty()

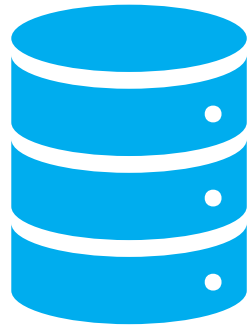
- Array – empty array

Consistency (1/2)

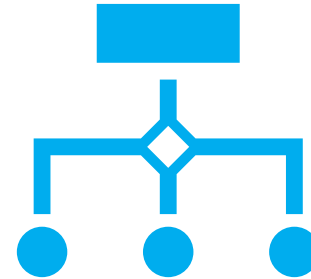
- Argument should be consistent
 - Order consistent
 - If argument size become unwieldy consider introducing a object parameter
- Refactoring will help consistency



Fit for Purpose



Do only one things and do it right (Single API call for One Scenario)



Understand the user and the customization that needed (Multiple API calls for One Scenario)

Restrained

- Justify the Public Method when creating new API
- Introduce protected API
- Convenience API is important
 - E.g `List.toString()`,
`list.add(Object)`
- Default position is make classes and public method final
- Start with private modifiers



Evolvable

Writing sample code with your API and discuss with real users

Sample codes should be FULL Sample

Review the sample codes

- Unclear

- Duplicate or redundant

- Abstraction is too high-level / low level

Quality Documentation

- Add annotation (@since, @link, @see)

- Include small code snippets as direct example

API Design in Azure

Case Study