Contents

[Fat Face Mulesoft Dev Environment 1](#_Toc500850495)

[Required Tools 1](#_Toc500850496)

[Setup and Configuration 1](#_Toc500850497)

[Development strategies / patterns 1](#_Toc500850498)

# Fat Face Mulesoft Dev Environment

## Required Tools

GitHub Client  
Maven

Anypoint Studio

## Setup and Configuration

GitHub repository is: Integrationhub

The Maven configuration file will need to be overwritten, Rupesh will provide me with this.   
get source code form GitHub need to import to Anypoint studio, file🡪maven from pom xml option,  
select (project object model) ‘pom.xml from Integrationhub repository ( directory?) for application or all ??  
  
https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html

<https://docs.mulesoft.com/anypoint-studio/v/6/importing-a-maven-project-into-studio>

Development strategies / patterns  
As the Mulesoft flows can be views as ‘functions’ we should not implement all functionality in a sing flow as this would be akin to writing al our code in one large function. To facilitate the use of multiple functions in an applications (Flows) we use the Mulesoft Flow Reference Component to control application execution flow. Doubling clicking on this component in Anypoint studio visual editor will take us to the flow(s) available. The Xml view of the component shows how the decision is made to direct to which flow, for instance POST or GET. In this way Flows can be designed to be reused (for common tasks, just as functions are) and called by other flows *‘Subflows’* are ideally suited for code reuse. Like a *‘Subflows’*, a ‘*Synchronous flow’* processes messages synchronously but does not inherit the calling flows Exception strategy etc…

The landing page or entry point flow in the application generally has the same name as the application.

All passwords (for connections etc.) are encrypted using AES any value beginning with ‘![‘ is encrypted. Global files hold environments (production, dev, CIT etc.) default is Dev.