->Using Any point studio to build, run, and test Mule applications

->Use a connector to connect to databases

->Use the graphical DataWeave editor to transform data

->Create RESTful interfaces for applications from RAML files

->Connect API interfaces to API implementations

---------------------------

Machine generated alternative text:
Mule event 
Mule message 
Attributes 
Payload 
Variables 

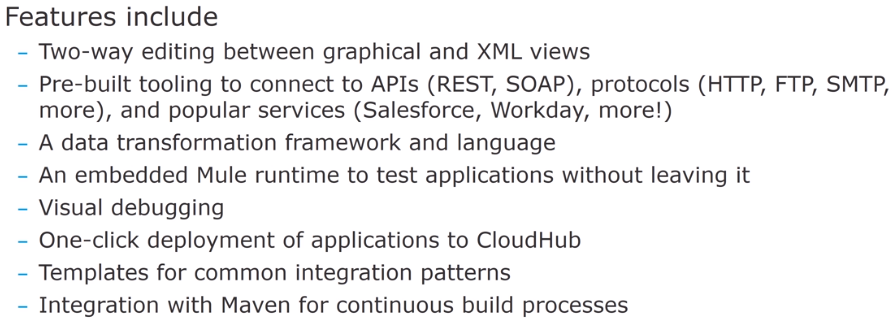
Payload-Contains data that is processed in the flow

'**Attributes**' would contain 'Query **Parameters**' and '**Headers**'

Variables can be defined within the flow and that can be processed by processors

Machine generated alternative text:
Anypoint Studio anatomy 
Package 
Explorer 
• Apt 
t. 1.9 
MuleSoft 
Mule 
Palette 
Canvas 
Console 

**Anypoint Studio** is based on **Eclipse**, a common Java integrated development environment



There are **two prospective** included with studio , one for **debugging** and another one for **design**

**Views->**Mule properties, Mule Debugger, MUnit, Munit Errors, Mule Palette…

Machine generated alternative text:
Anatomy of a flow: Visual 
getFIightsF10W 
Event source 
GET 
ights 
All flights 
LOG 
JSON to JSON 
Error handling 
Event processors 
On Error propagate 
Set Payload 
Error handling 
var 
Set Variable 

Source, Exception handling sections optional

Processors section mandatory

Machine generated alternative text:
-L OCT 
Logger 
JSON to SON 
GET "lights 
Err« handäng 
All 'lights 
138 
16 
On Error Propagate 19 • 
•:flow name—egetF7ightsFlO"" doc: id— "fdIb716S-S456-476a-818f-4b25Ø16b4a26" 
cheep: listener doc: name— "GET 'flights" doc: id— "9?cb8føf-2cb3-48ø9-ab61- 
<db: select doc:name— flights" doc: id— "13978339-aZ63-451Z-9b6b-114f2 
*SELECT • 
ddb : selects 
eee:transform doc: to 'SON" 
«ee : message 
«ee:set-payload 2.ø 
output application/json 
21 
payload 
23 
Set Payload 
25 
26 
27. 
3ø 
32 
dee : me s sag 
: form* 
-aogger• doc: name— "Logger" doc: id— "5b1628cf-ec96-4beb-81d6- 
€on-error-pr•opagate enableNOti logException— "true" 
«set-payload value- "'CpayloadJ" doc:name— "Set Payload" doc: id-• 
eset-variable value— "test" doc: Variable" doc: id— "f4Øe 
</on -e r ror- 
</error -handl 

By default, projects include **Core, HTTP, Sockets , Add Modules (Connectors), Search in Exchange**

**Core Modules**->Primary **components** to create flows

Modules used to interact with HTTP, TCP and EDP protocols

To add more functionality, We can add more modules ,either from ones that are installed in studio or by downloading from Exchange.

Anypoint studio comes with an **embedded Mule runtime** to test applications without leaving it.

**Console** outputs **application logs** and **information**

Can automate testing of Mule applications using MUnit.

Runtime->Mule Server 4.2.2 EE

Anypoint studio version ->7.4.2

Note: It is recommended that for greater product stability and to avoid excessive memory consumption you set the mule.apikit.parser VM argument for this new project as outlined in the ***Check for course updates and/or issues*** section of this course.

'Set Payload' component->To set payload

Database connector can connect to almost any data base engine for which you have a driver

Data retried from d/b is in JAVA (Array object) format, need transformation to convert to other format (either XML/JSON) using **DataWeave 2.0** transformation

->APIKit is an open-source toolkit that includes an Anypoint Studio plugin.

-> The Anypoint Studio APIKit plugin can generate an interface automatically from a RAML API definition (For new or existing projects).

->It generates main routing flow and flows for each of the API resource/ method pairs. Will add processors to the resource flows to hook up to your backend logic.

->'Flow Reference' component is used to call from one flow to another flow