

Make.Com Notes

Day 0

- Common tools
 - Data Bases
 - CRM
 - Email / Marketing automation
 - E-Commerce
 - CRM
 - Project management
 - Office
 - Telephony

Day 1

- JSON Essentials

- Array ex: ["sheet", "word", 234]
 - Collection ex: { "number": 20, "name": "A/er",
"key": "value" } ~ Payment Data ~ "05/01/2021"
 - object: a collection of name/value pairs
- Formatting:
- string in double quote
 - NOT in quote.
 - backslash for escape special characters.

Day 2

- Variables
- Functions

Tip: can check JSON bundle to get the target value
when creating variables.

- * General function.
 - () switch
 - get omit
 - if pick.
 - if empty
- * Date / time Functions
* Math Functions
* Text Functions
* Functions working with Arrays

Day 3

- Connection.

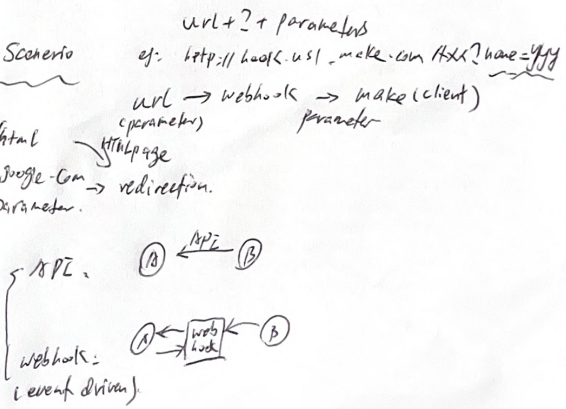
- webhook one webhook per Scenario

Webhook response

{ content-type: text/html
location: https://google.com → redirection.

HTTP { Get - url? parameter.
post - JSON

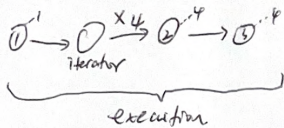
API vs. webhook.



Day 4 Execution & cycles

Execution { "Run" button.
scheduled trigger.
variable lifetime. (cycle / ^{one} execution / ^{one} Node).

iterator → cycle
* Many cycles in one execution



x1 after iterator - one cycle.

x4 means 4 bundles, 4 cycles

Day 4 iterator and aggregator

iterator - input: Array
- output: Bundles of items in Array

or choose variable
* Create Array directly or use Map function

eg: `split(test1, test2, 123, 1)`
⇒ `[test1, test2, 123]`

Aggregation * incorrect sources can lead to inaccessible variables outside aggregation.
eg: Aggregating email attachments using current sources to ensure data accessible.

* Variable inside aggregation sequence are NOT available outside of it.

Tips: Group array by parameters such as file name, using functions like split and Get.

Aggregator: { Array Aggregator
Text Aggregator
Table Aggregator
Numeric Aggregator

* rmb for preventing errors by stopping process if aggregation result is empty

Day 5 Complex Arrays

API → Arrays (names, email, phone number)
get data

- Map Array₁ map Array₂ New JSON
 phone number phone number merged field

* Efficiency - Avoid multiple API calls by merging arrays in a single operation.
(API limits) ↓
reduce potential errors / optimize operation cost.

Day 7 HTTP Request, HTTP Module.

- Most module in make use APIs for external communication
- Custom API calls can be made using HTTP modules

Basic API Request structure

- URL
 es. bearer token & URL require parameters encoded
 es: space \rightarrow %20
 @ \rightarrow %40
- Header: for authorization / content-type
- Body: contain data to be sent in POST Request

Request Methods:

- GET (Retrieve Data)
- POST (Send data / create or update resources)

Responses

- successful - 200
- redirection - 300 / 3XX
- client error - 400 / 4XX
- server error - 500 / 5XX

Custom Request Types:

- multi-part requests (text in request body) make & structure data
- Custom text
 - set request body to custom
 - set header (separator / put text in request content)

Authorization setup:

- username + password - basic authorization (base64 - encoded)

Request Timeout

Default - 40s / Max - 5 mins per call

* Reject connection if certificate is incorrect \rightarrow Certificate validate / fail

HTTP Modules

- GET Files
- Resolve Target URL

Custom API calls

specifying the type of data (Text, JSON, Binary)

Get

URL + ? + parameters

POST

method + headers + Body

Request Body Types

- URL encoded forms
- multi-part forms

Always

set content-type header

JSON

* API Body

JSON format

JSON \rightarrow Body

JSON parsing \rightarrow structure data

* website \rightarrow cookies \rightarrow authorization \rightarrow header

certificate-based \rightarrow Services call pages

API call \rightarrow applications (Jira) spreadsheet