CREATING A MAPPING TEMPLATE

Hello! folks its awesome to have you here working towards understanding how to create a mapping template. Before we get to know the implementation part lets understand why we need a model.

TOPICS:

* [Why to implement a Mapping Template](#_WHY…?)
* [Velocity Template Language](#_VTL)
* [JSON Path](#_JSON_PATH)
* [Creating A Mapping Template](#_CREATING_A_MAPPING)

# **WHY…?**

AWS uses it to map the payload from a method request to the corresponding integration request and from an integration response to the corresponding method response. In simple terms it is used to transform an incoming payload into a different format.

Diagram, timeline

Description automatically generated

API Gateway allows you to define input mapping templates, for mapping the incoming request from a client to a server format, and output mapping templates, for mapping the outgoing response from the server to a client format.

The mappings are defined using the [Velocity Template Language](http://velocity.apache.org/engine/devel/vtl-reference.html) combined with [JSONPath](http://goessner.net/articles/JsonPath/) expressions.

# **VTL**

Velocity is a Java-based templating engine that uses a template language to reference objects in Java code. Velocity can be used to generate web XML, HTML or SQL from templates, for example. Here, we will use it to define the mapping between an input and output model.

A Velocity Template Language (VTL) template is composed of statements which begin with the # character and a followed by a directive. When the VTL engine is run on your template, it searches for all # characters to find any that match a VTL directive. For example, the following VTL statement contains the set directive.

#set($a = "Velocity")

VTL contains a number of directives that can be used to setting values, running conditional statements, and looping, among others.

The second major component of VTL is references, which begin with the $ character. More generally, references begin with $ and are used to get or set something. Directives begin with # and are used to do something.

We can use the statement above to define an HTML template that sets a reference to a value and retrieves that reference.

<html>

<body>

#set($foo = "Velocity" )

Hello $foo World!

</body>

</html>

For more on VTL, consult the [user guide](http://velocity.apache.org/engine/devel/user-guide.html#velocity-template-language-vtl-an-introduction) or the [language reference](http://velocity.apache.org/engine/devel/vtl-reference.html).

# **JSON\_PATH**

In API Gateway mapping templates, the input to your Velocity template is defined by the reference variable $input. The API Gateway runtime provides this value to your template for use. To access particular portions of the input JSON document within the template, you use JSONPath.

JSONPath provides an XPath-like method of referring to a JSON structure. IN JSONPath, the abstract name $ refers to the root-level object. From this root-level object, you can use dot-notation to select more restrictive portions of the document. For example, to reference the playerId of the player model, you can either of the jsonpath expression.

#['playerId’] or $.playerId

# **CREATING A MAPPING TEMPLATE**

The Initial payload that needs to be transformed is

{

"studentDetails": {

        "name": "Griezmann",

        "course": "Mechanical",

        "nationality": "Canadian",

        "phoneNumber": "98979798687"

    },

    "tuitionFee": "$1000"

}

After creating an API, go to Integration Request and configure the method in this select **mapping template** follow it up by providing the second option in **Request body passthrough**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

{

“TableName”: “studentData\_DDB”,

“Item”: {

“studentName”: “Griezmann”,

“course”: “Mechanical", // Extraction from Payload

“registrationID”: “45678” // Extraction from a queryParam

}

This marks the end of creation of mapping template and attaching it to transform the incoming payload [request information]. **Happy Learning!!**