Step Functions Tutorial

AWS Step Functions is fairly straightforward. The following are some brief notes that will help you create State Machine and integrate it with other web services.

Create a State Machine

- Click on State Machines and select "Blank"
- Add a Task to the workflow
- Click on "Config" (top menu)
    - Select the IAM role (e.g. LabRole) and name your state machine
    - You should turn on Logging while testing your State Machine. This will write all events to CloudWatch and will help you diagnose issues.

Configure API Gateway to call your State Machine

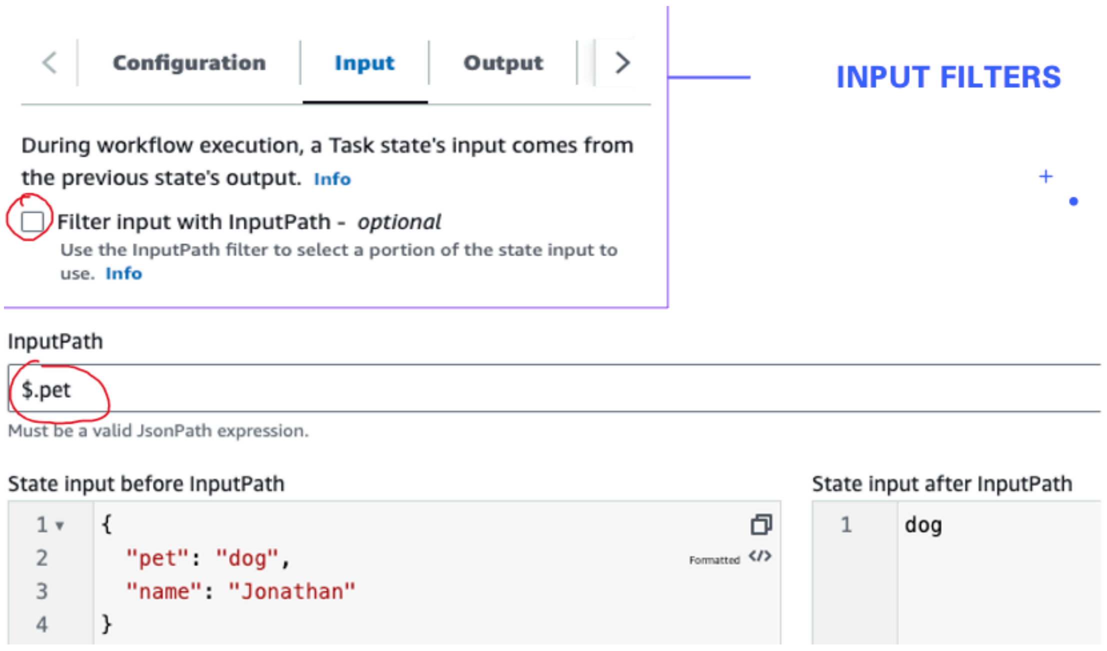
- Create a new API Gateway. (Optionally create a new Resource.)
- Create a new Method for "POST".
    - For Integration type, select "AWS Service"
    - Select your region and "Step Functions" as the service
    - Select the HTTP method that matches your Gateway Resource Method
    - Action Type = "use action name"
    - Action name = "StartExecution"
    - Paste the IAM Role ARN. (For the Learner Lab, this will be "LabRole".)
- Edit your Integration Request for your new Method.
    - Expand "Mapping Templates" and past the following template body for "application/json".
        - Replace "account-number" with your AWS account number.
        - Replace "MyStateMachine" with the name of your AWS Step Functions state machine.

```
{
  "input": "$util.escapeJavaScript($input.json('$'))",
  "stateMachineArn": "arn:aws:states:us-east-1:account-number:stateMachine:MyStateMachine"
}
```

- Don't forget to **deploy the API** to a new stage ("prod")

Input and Output Filters

The following images show how to create the filters so that the input and output flows in a way that is easy to parse between steps. Whenever you are logged into AWS, you can use Data Flow Simulator to experiment with your own data models.



During workflow execution, a Task state's input comes from the previous state's output. Info

[ ] Filter input with InputPath - *optional*
Use the InputPath filter to select a portion of the state input to use. Info

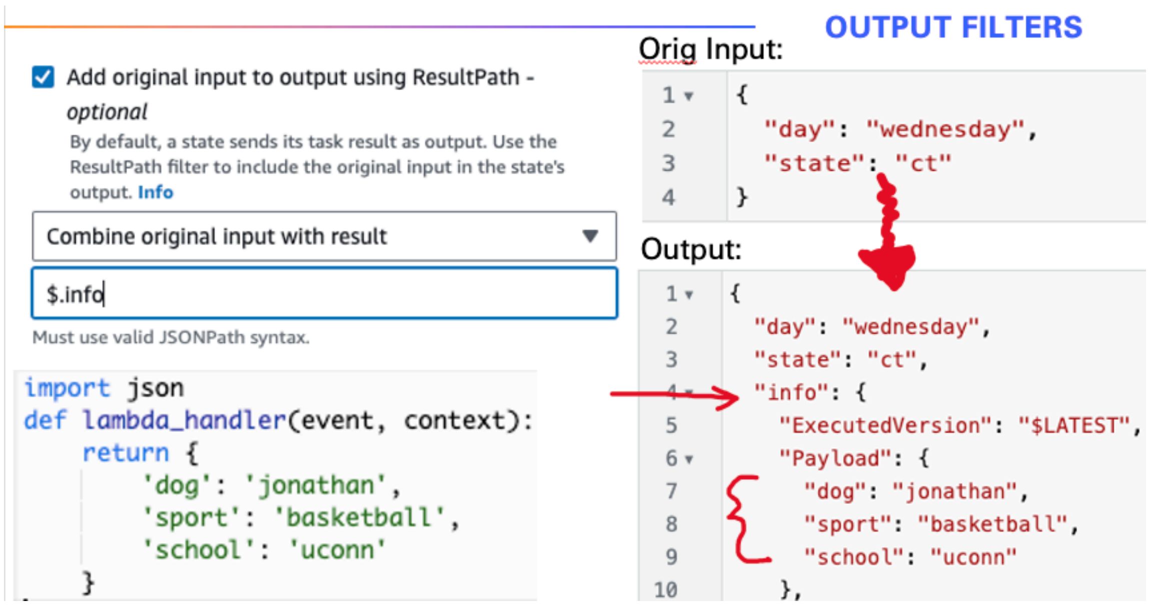
InputPath: `$.pet`
Must be a valid JsonPath expression.

State input before InputPath

```
{
  "pet": "dog",
  "name": "Jonathan"
}
```

State input after InputPath

```
dog
```



[✓] Add original input to output using ResultPath - *optional*
By default, a state sends its task result as output. Use the ResultPath filter to include the original input in the state's output. Info

Combine original input with result

`$.info`

Must use valid JSONPath syntax.

```python
import json
def lambda_handler(event, context):
    return {
        'dog': 'jonathan',
        'sport': 'basketball',
        'school': 'uconn'
    }
```

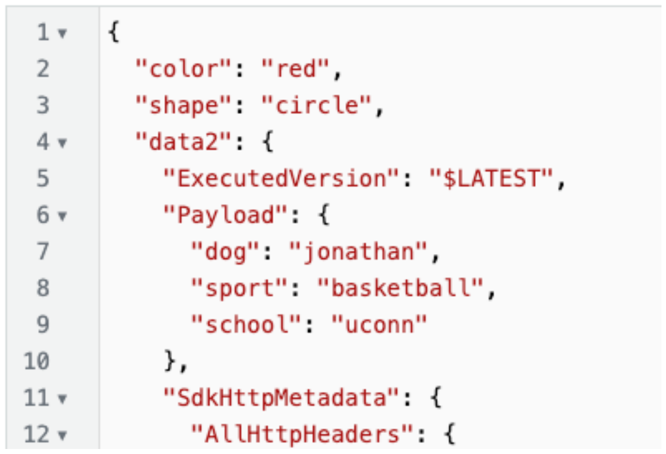
Orig Input:

```
{
  "day": "wednesday",
  "state": "ct"
}
```

Output:

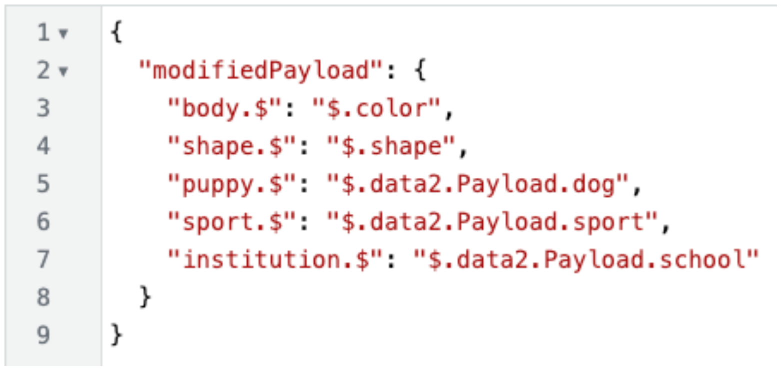
```
{
  "day": "wednesday",
  "state": "ct",
  "info": {
    "ExecutedVersion": "$LATEST",
    "Payload": {
      "dog": "jonathan",
      "sport": "basketball",
      "school": "uconn"
    },
```

Task result before ResultSelector



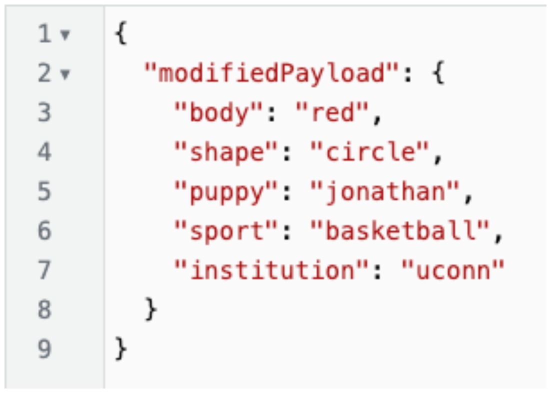
```
{
  "color": "red",
  "shape": "circle",
  "data2": {
    "ExecutedVersion": "$LATEST",
    "Payload": {
      "dog": "jonathan",
      "sport": "basketball",
      "school": "uconn"
    },
    "SdkHttpMetadata": {
      "AllHttpHeaders": {
```

ResultSelector



```
{
  "modifiedPayload": {
    "body.$": "$.color",
    "shape.$": "$.shape",
    "puppy.$": "$.data2.Payload.dog",
    "sport.$": "$.data2.Payload.sport",
    "institution.$": "$.data2.Payload.school"
  }
}
```

Task result after ResultSelector



```
{
  "modifiedPayload": {
    "body": "red",
    "shape": "circle",
    "puppy": "jonathan",
    "sport": "basketball",
    "institution": "uconn"
  }
}
```

Adding a Manual Approval Step

- SNS
  - Create a new Topic
    - Access Policy - Anyone can publish (or specify a specific AWS Resource in the policy)

    - Create a subscription
        - Protocol = Email
        - Endpoint = <your email address>
- Step Functions
    - Create an Activity and give it a name. It is basically a placeholder to get the workflow to wait for the callback. We don't need an Activity Worker to poll (as most tutorials will state) because are also sending message to an SNS Topic which will have a SendEmail subscriber.
    - Add the "Run Activity" state to your workflow and select your new activity.
    - Add a "SNS: Publish" state to your workflow.
        - Select your SNS Topic
- API Gateway
    - Create 2 new resource: /pass and /fail
    - Create a Method for each.
    - For each, select "AWS Service" and "Step Functions"
        - Select us-east-1, POST
        - Set the Execution role to be the ARN for LabRole (obtain the ARN from IAM)
    - For the "/pass" method, set the Action Name = SendTaskSuccess
    - For the "/fail" method, set the Action Name = SendTaskFailure
    - Edit the "Integration Request"
        - Add a Mapping Template for "application/json"

```
{
   "cause": "Reject link was clicked.",
   "error": "Rejected",
   "taskToken": "$input.params('taskToken')"
}

{
   "output": "\"Approve link was clicked.\"",
   "taskToken": "$input.params('taskToken')"
}
```