Holiday Partner Services - Partner Paging Support

This page details the high level changes required to support paging partner results in Holiday Partner Services. This implementation strives to retain the partner agnostic nature of the Holiday Partner Service implementation.

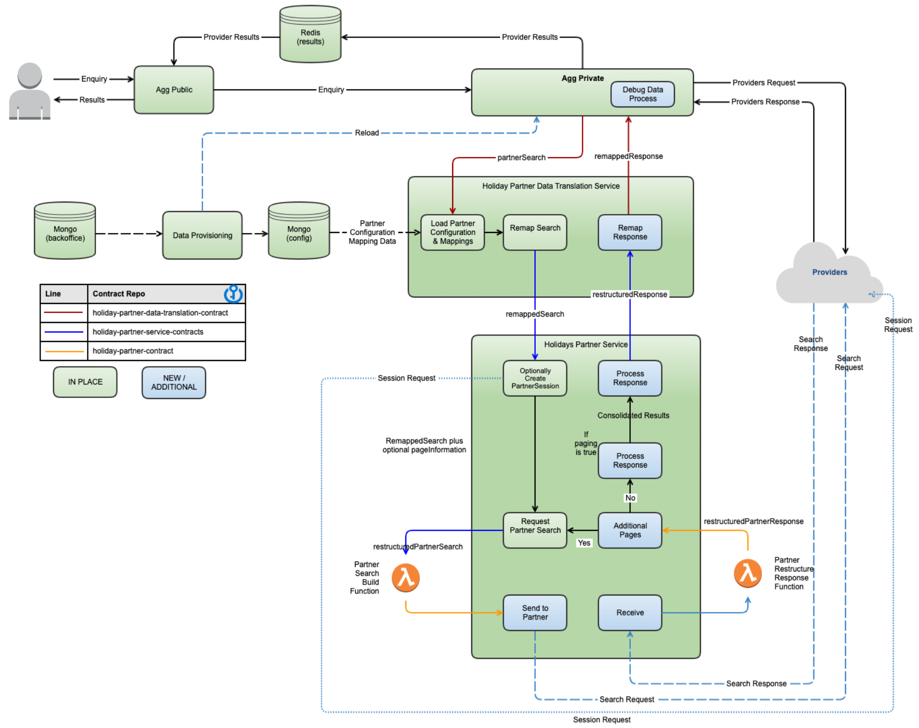
Holiday Partner service will honour instructions to perform a subsequent search request, will provide a mechanism for paging data to be passed from one invocation of a lambda function to the next invocation and will consolidate and finalise the results to maintain the existing contract with the Data Translation Service. Note although the data portion of the contract is maintained there is a change to the metaData portion of the contract.

The lambda response functions will need to determine the correct mechanic for managing a specific partner paging implementation. The response function will populate return a flag indicating if additional pages are needed, as a flow control field, along with pagingInformation containing dynamic state data. When the is another page call HoPS will make an additional call to the lambda search function, passing pageInformation, to start the next partner call iteration. Once all iterations have completed or we have hit the maximum time allowed for paging HoPS will returned the consolidated set of results.

Stories

- Contract Changes
- HoPS paging implementation
- Travelsoon Lambda function upgrade (Conform to new contract)
- EasyJet Lambda Paging Implementation
- Data Translation Service Implementation
- Aggregation Write multiple partner request/responses for debug
- Investigate/Decide on release of partner lambda changes (alias) (Colin)

  Optional/Stretch
- Add pagesize to ProviderChannelConfiguration
- DTS to extract and pass pagesize
- contract changes to support new pagesize (request DTS/HoPS/Lambda)
- Hops to pass pagesize to lambda
- Travelsoon lambda function upgrade
- EasyJet implement use of pagesize



High Level Requirements

- Service code to be developed and ran under JDK 11
- Lambda code written under JDK 11
- Implementation should provide a partner agnostic paging solution in Holiday Partner Service
- Holiday Partner Service provides passing paging state between partner http requests
- Individual Lambda implementations control the paging criteria/process
- Holiday Partner Service to impose a strict time limit (25 seconds) before returning all available results if paging is incomplete.

Solution outline

- Partner response lambda determines the URL for the next page of results
- Partner response lambda provides a paging information block to persist state to next call
- HoPS will check for a next page url, calling it if exists and passes the new response plus the last return paging information block to the response lambda
- During paging HoPS consolidates the partial results and issues the next page URL call
- When there is no URL HoPS will finalise the results and build a standard results response

Stretch Requirements

- Add page size to provider (Partner) configuration in the backoffice. This data would be added by DTS and would extend the request contracts down to and including the two lambda function contracts.

Details

Holiday-Partner-Contract Changes (equivalent changes would need to be made to the holiday-partner-service-contract and holiday-partner-data-translation-service-contract).

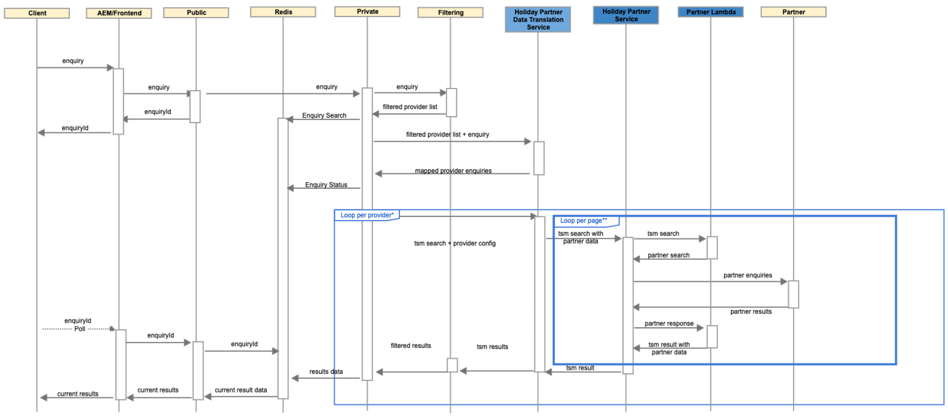
```
restructuredPartnerSearch
request {
      ...
      partner: {
         ...
         "pagingInformation": ""
    }
    ...
}
```

```
lambdaResponse
{
    "restructuredPartnerResponses": [ ],
    "partnerCallMetaData": {
        "paging": {
            "information": "lambda managed string",                   // Optional
            "nextUrl": "lambda defined URL to retrieve next page"    // Optional: if the element is empty or
doesn't exist no paging
        },
        "valid": 7,    // Aggregate across all pages
        "rejected": 3,  // Aggregate across all pages
        "partnerRequest": "captured request",
        "partnerResponse": "captured response"
    }
}
```

This replaces the partner request and response with a partner call section and adds the two paging fields.

Sequence Diagram for calls using holiday partner service with paging

Sequence Diagram



* The provider list is actually a set of, potentially, multiple calls based on provider configuration. This is to support examples where a single search for us is a multiple set of searches on the provider site: e.g. if the provider doesn't support days either side on a single call.
** Paging is determined by each individual partner implementation

Simple Examples:

Partner includes a simple "more data" field with a page-number parameter in the URL

The search lambda function would create the search for page 0/1 or without page information (partner implementation specific detail)

The response lambda function would check the status of the "more data" field and build the next URL. In this case the pagingInformation could just be the last page number as an int. In this case the function will take the page number, generate the new URL and save the new page number into paging information.

Partner provides an initial "digest" response detailing the number of pages

The search lambda function would create the standard partner search.

The response lambda would process the digest and, probably, generate a POJO to hold the on-going paging information and generate the first result page request URL. It would then populate the response with the new URL and serialise the POJO into the pagingInformation parameter.
On the subsequent calls the function would deserialise the paging information, update its page information and prepare the response.