HTTP Server Source/Destination Connector

Source

Config:

Name	Description	Required	Example	Default value
Port	http server will run in this port	False	8888	3000

Implementation:

Source `open` method starts HTTP server on `port` in separate goroutine. HTTP server has method for listening post requests. When server catches post request send request to special chan, if get error send error to errors chan.

```
func (h *HTTPServer) listen(w http.ResponseWriter, r *http.Request) {
        if r.Method != http.MethodPost {
                http.Error(w, "Method is not supported.", http.
StatusNotFound)
                return
        }

        rawBody, err := ioutil.ReadAll(r.Body)
        if err != nil {
                h.errors <- fmt.Errorf("read body: %w", err)

                http.Error(w, err.Error(), http.StatusBadRequest)
                return
        }

        h.data <- rawBody

        w.WriteHeader(http.StatusCreated)

        return
}
```

HTTP server has method to GetRecord, where on loop we listening chan with body requests and transform it to `Record` and close loop when get context cancel event.

```
func (h *HTTPServer) GetRecord(ctx context.Context) (sdk.Record, error)
{
	for {
		select {
		case err := <-h.errors:
			return sdk.Record{}, fmt.Errorf("get record: %
w", err)
		case data := <-h.data:
			pos, err := generatePosition()
			if err != nil {
				return sdk.Record{}, err
			}

			return sdk.Record{
				Position:  pos,
				CreatedAt: time.Now(),
				Payload:   data,
			}, nil
		case <-ctx.Done():
			return sdk.Record{}, ctx.Err()
		}
	}
}
```

For generate position we will use uuid, with `"github.com/google/uuid"` library.
Also http server will have method to shutdown server and closes channel.

```
// Stop - shutdown server and close channels.
func (h *HTTPServer) Stop(ctx context.Context) error {
	err := h.server.Shutdown(ctx)
	if err != nil {
		return fmt.Errorf("shutdown server: %w", err)
	}

	close(h.data)
	close(h.errors)

	return nil
}
```

Issues/Problems/Questions/Suggestions.

N/A

Destination

Config:

Name	Description	Required	Example
URL	URL which will be using for send post requests	true	`http://localhost:8888/data`
HTTP header	http header can be using for adding token	false	`Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==`

Implementation:

Source method `open` will ping `url` from config. Source method `write` will get record transform to post request and send post request to `url`, and check http code response. If it HTTP Status 201 (Created) or HTTP Status 200 OK, other way get error. We are planning to do retries (3 times) if request failed. Method `teardown` close connection.

Issues/Problems/Questions/Suggestions.

N/A