Manage Invoices

# some references

* https://www.invoiced.com/resources/docs/api/#introduction could be intresting to look at as it is an invoicing api i am not reall copying this but gt some ideas from it.
* https://livebook.manning.com/book/api-design-patterns/chapter-9/30

# requirements

*a user should be able to:*

* create an invoice record
* email that invoice either from their own domain or from the email client on the device they are using the application on.
* void an invoice
* see what job records have:
  + had an invoice created
  + have had an invoice sent
  + have had an invoice sent but invalidated
* a way to see all invoices
* a way to see invoices for a certain booking as some bookings may have multiple invoices due to marking

# ui features overview

* a list of job records showing the current invoice state:
  + not created, created, sent, invalidated
* a page that can be opened from the booking records list to see all details for the booking, any invoice records and perform invoice actions:
  + create invoice
  + email invoice
  + invalidate
* a list of invoice records to summarise information about invoices e.g bill, paid and calculate summary statistics.

2 of the features create invoice and email invoice require a bit more investigation it makes sense to look at how this is done on the api before considering how these features will work on the front end.

# api design

## invoice model

the invoice model Is created from data that can also be found on the job model. However the invoice model will have more strict validation. The job model has rather loose validation as it can be created at an early stage of the booking however when we are creating the invoice we there is more information that is required.

## invoice model implementation note

move the object which sets up the job model up to a shared file jobObj.js job and invoice then both import this file and make some modifications to it, it will mainly be invoice.js that changes this as it is imposing more strict validation.

a way of marking the job.js model as validated for invoice so that it can easily be checked -use a virtual property on job.js which uses the invoice.js validation to check if it is validated as an invoice and sets readyForInvoice to true if it passes.

## Create invoice endpoint

The invoice is really a snapshot of a job record, however if we made a mistake on an invoice we do not amend the invoice we void it (should I have an update endpoint or a specific voidInvoice endpoint as this is all we should be able to modify), so we would have to update the job record then create a new invoice record. We have the option to get the data for this snapshot from the database or sent from the ui after having been fetched from the database. Discussed later in ui section as I have currently decided to send from ui. However should perhaps be making a check when sent over to make sure it is the same as what Is on the db. Not sure of the best practice to deal with this.

## Email Invoice endpoint

This is an endpoint that has a side effect therefore it makes sense to use a custom method see:

* https://livebook.manning.com/book/api-design-patterns/chapter-9/ JJ -custom method for sending email.
* https://github.com/expressjs/express/issues/3857 - putting a colon in request

problem: needs and email account to send from

possible solution: send from own domain

user probably wants to send from there own domain they could create an email address on there domain e.g. [invoice@mydomain.com](mailto:invoice@mydomain.com) to send from this would have its own email address

drawback:

maybe don’t want to provide a credentials but basic email domains don’t have

possible solution: use google account with oauth 2

secure

drawback:

businesses generally want to use there own domain

currently favour: send from own domain we need a email address model