Uma imagem com mesa

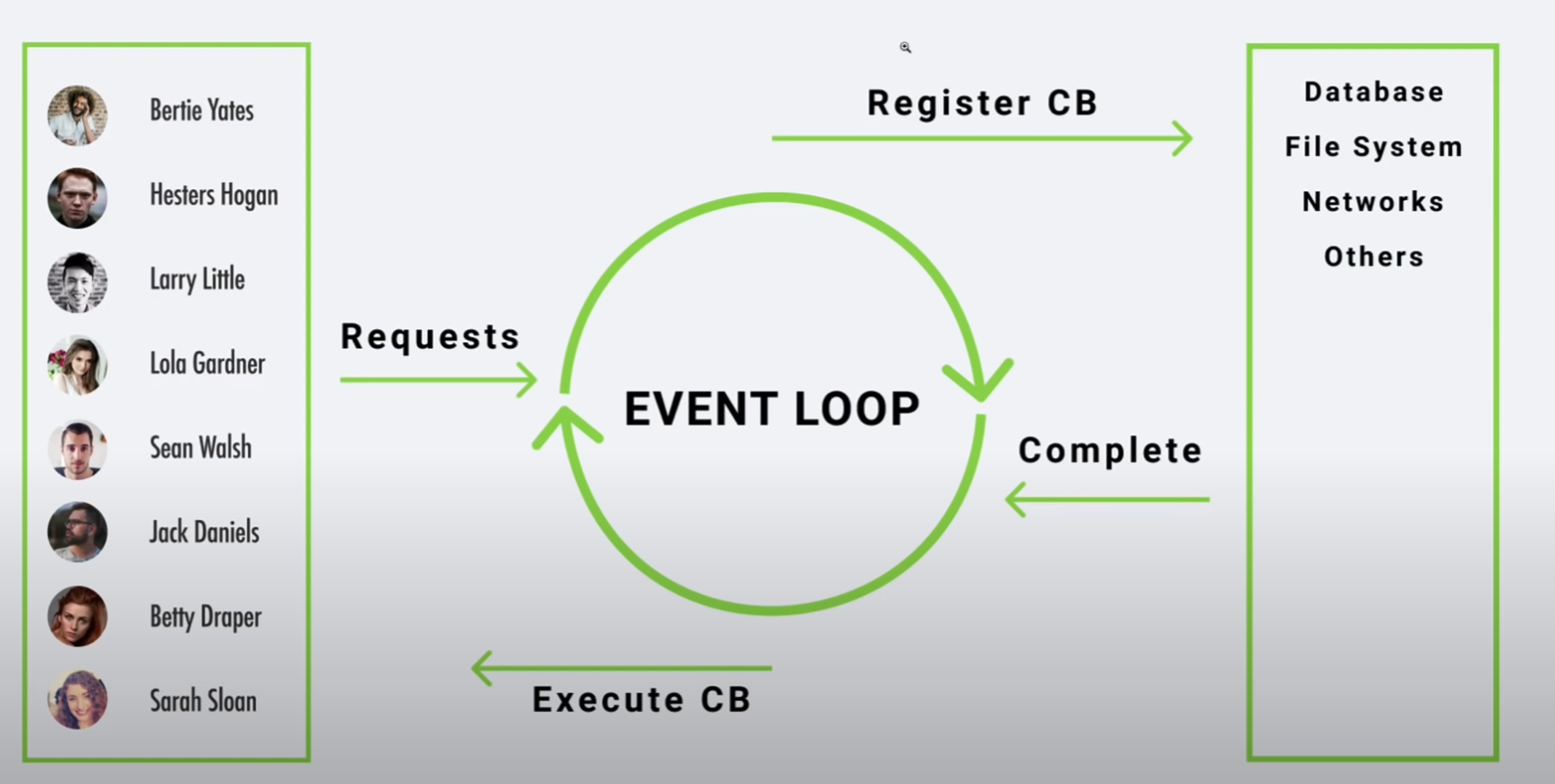
Descrição gerada automaticamente

Course repository => <https://github.com/john-smilga/node-express-course>

Course Video => https://www.youtube.com/watch?v=Oe421EPjeBE&t=28831s

CommonJS, every file is module (by default)

Modules - Encapsulated Code (only share minimum)



To the right we have the operations that the event loop regists the call back function, in order

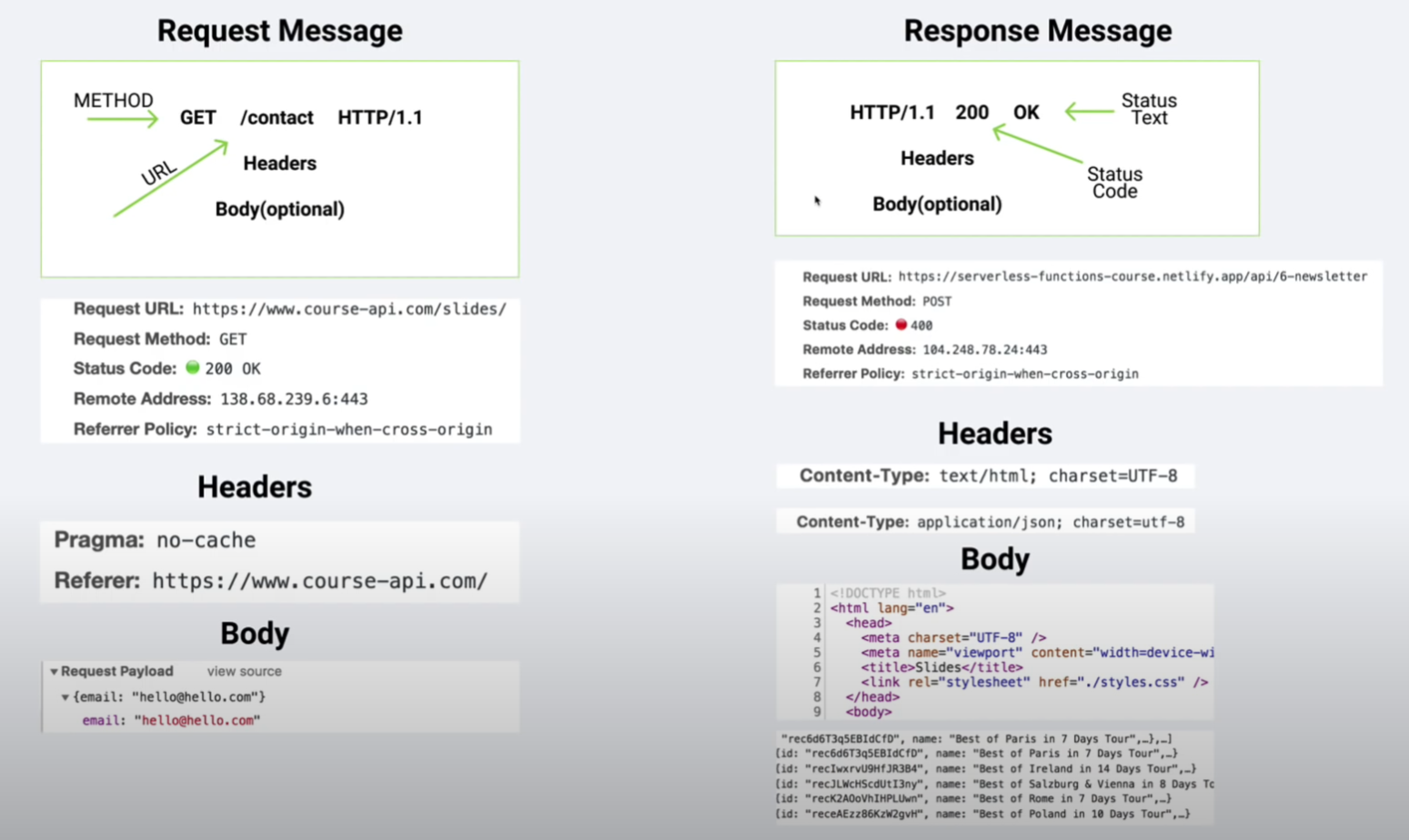
to proceed to other tasks in the event loop.

Uma imagem com texto

Descrição gerada automaticamente

Handle event in node.js (server side)





Uma imagem com mesa

Descrição gerada automaticamente

Node express course Github => <https://github.com/john-smilga/node-express-course>

Uma imagem com texto

Descrição gerada automaticamente

API – interface em http que retorna objetos em formato JSON(javascript object notation) que

trata (res.JSON()) de maior parte das coisas. Por exemplo define logo o content-type e stringify a nossa informação

SSR – server side rendering – templates para enviar todo o js, css, html por nós mesmos (res.render())

Uma imagem com texto, captura de ecrã, monitor, preto

Descrição gerada automaticamente

Tudo o que vêm nos parâmetros do request é uma String, ter a atenção depois no uso destes valores durante a pesquisa de informação na aplicação.

Uma imagem com texto

Descrição gerada automaticamente

Temos neste caso um exemplo de uma possível implementação.

Uma imagem com texto

Descrição gerada automaticamente

É necessário verificar sempre situações em que o pedido do cliente não têm resposta possível, ou que a informação não consta na aplicação entre outras situações…..

Uma imagem com texto

Descrição gerada automaticamente

Neste exemplo retornamos o produtos todos caso a route query seja preenchida sem parâmetros, depois verificamos todos os parâmetros se existentes e fazemos a devida filtragem ou procura (entre outras). **Não esquecer situação em que não é encontrado nenhum resultado para o pedido.**

**Atenção** Só podemos mandar uma resposta por cada request, atenção ao retorno des res.send|res.json, caso o código não o disponha, o código a seguir acaba por ser executado. Fornecendo o seguinte erro (‘Error [ERR\_HTTP\_HEADERS\_SENT]: Cannot set headers after they are sent to the client’)

Middlewares

Uma imagem com texto

Descrição gerada automaticamente

Express fornece já o req, res e next aos middlewares ! Devemos sempre após a execução do middleware chamar a próxima função, a não ser que no middleware envio a res (resposta do servidor), contudo não é adequado.

Uma imagem com texto

Descrição gerada automaticamente



Neste caso chamamos next para voltar para o nosso app.get method.

Como poderemos associar o middleware com o express internamente ?

Uma imagem com texto

Descrição gerada automaticamente

O middleware logger e authorize é evocado ‘automaticamente’ em qualquer rota escolhida.

Como associar o middleware a paths específicos ?

Uma imagem com texto

Descrição gerada automaticamente

Como temos ilustrado no exemplo em cima, associamos o prefixo dos paths para os quais queremos executar o middleware.