

WEB PROGRAMMING

Practice M6

1. Browser: Safari, Version 14.0
2. Operating system and Computer Architecture:

System Software Overview:

System Version:	macOS 10.15.7 (19H2)
Kernel Version:	Darwin 19.6.0

Hardware Overview:

Model Name:	MacBook Air
Model Identifier:	MacBookAir8,2
Processor Name:	Dual-Core Intel Core i5
Processor Speed:	1.6 GHz
Number of Processors:	1
Total Number of Cores:	2
L2 Cache (per Core):	256 KB
L3 Cache:	4 MB
Memory:	8 GB

3. How REST helps your web application:
REST's simplicity (simpler data formats. Such as JSON, are typically used) and reduction of overhead are its main benefits.
4. Additional features:
 - a. Add an `updateContact()` function to change an existing contact.

```
//Additional function
public updateContact(inID: string, inContact:IContact): Promise<string>{
  console.log("Contacts.Worker.updateContact()", inID);

  return new Promise((inResolve, inReject) => {
    this.db.update({ _id : inID },inContact,{
      {inError: Error|null, inNumUpdated: number,inAffectedDocuments: any, inUpsert:boolean} => {
        if (inError) {
          console.log("Contacts.Worker.deleteContact(): Error", inError);
          inReject(inError);
        } else {
          console.log("Contacts.Worker.deleteContact(): Ok", inNumUpdated);
          inResolve();
        }
      }
    });
  });
}
```

- b. Set HTTP response code by create a separate file "HTTPStatusCode"

```

//Get a list of mailboxes
app.get("/mailboxes",
  async (inRequest: Request, inResponse: Response) => {
    console.log("GET /mailboxes (1)");
    try {
      const imapWorker: IMAP.Worker = new IMAP.Worker(serverInfo);
      const mailboxes: IMAP.IMailbox[] = await imapWorker.listMailboxes(); //listMailboxes: capture the a
      console.log("GET /mailboxes (1): OK", mailboxes);
      inResponse.json(mailboxes).status(code.default.GetBoxListS_code);
    } catch (inError) {
      console.log("GET /mailboxes (1): Error", inError);
      inResponse.send("error").status(code.default.GetBoxListF_code);
    }
  }
);

//182
//Get a list of messages in a specific mailbox
app.get("/mailboxes/:mailbox",
  async (inRequest: Request, inResponse: Response) => {
    console.log("GET /mailboxes (2)", inRequest.params.mailbox);
    try {
      const imapWorker: IMAP.Worker = new IMAP.Worker(serverInfo);
      const messages: IMAP.IMessage[] = await imapWorker.listMessages({ //listMessages: get an array of IAP.IMe
        mailbox : inRequest.params.mailbox
      });
      console.log("GET /mailboxes (2): OK", messages);
      inResponse.json(messages).status(code.default.GetMessageListS_code);
    } catch (inError) {
      console.log("GET /mailboxes (2): Error", inError);
      inResponse.send("error").status(code.default.GetMessageListF_code);
    }
  }
);

```